ANALYSIS OF THE RESEARCH PAPER

Work done by KARTHICKTHASAN S(CB.SC.P2CSE24006)

Reference : AutoGen Studio: A No-Code Developer Tool for Building and Debugging Multi-Agent Systems

Authors: Victor Dibia, Jingya Chen, Gagan Bansal, Suff Syed, Adam Fourney, Erkang Zhu, Chi Wang, Saleema Amershi from Microsoft Research

My research methodology assignment paper of choice is AUTOGEN STUDIO: A No-Code Developer Tool for Building and Debugging Multi-Agent Systems. While looking for a study paper, I visited several companies' websites to see their work. I finally found this paper from Microsoft. While searching for papers on artificial intelligence, I discovered that the majority of the articles are on huge language models, neural networks, and so on, and I have passed about 10 research papers.Why this paper was chosen is because it provides context for how LLMs might interact with existing tools and communicate with one another to make the process much more efficient and it is much more different from all other paper.

During my first pass I started reading the abstract of the paper that is all about building the communication between the large language model and tools with the help of the open source framework called AutoGen developed by Microsoft.

Microsoft emphasizes that they build based on the majorly three problems that prototyping of the Agent is not easy , debugging the issues happening between the ai model and tools were not visible, already existed frameworks like LiDa , langchain have limitation because of the defined pipeline which is nothing but the what is really happening between them, because of the defined pipeline, they said it cannot flex like what they wish. In addition, the Generative models are limited in performance to certain tasks. And also that there are similar products available in the market called Teamweaver products where the developer has to give a large number of parameters by defining the model, tools to the agent, and complex interactions.

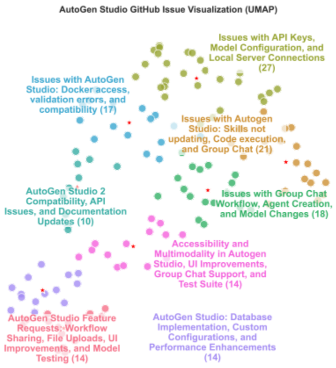
To the part of the hypothesis, they just generalized how they defined the problem and solved it by proposing the framework which can handle all the major problems that occurred between the model and the tools . They wanted to simplify the process by creating the platform to solve the major issues happening between the model and the tool with the developer tools like debugger, along with no code platform to minimize the process of making the prototype in the minimal period of time in addition to that they wanted to created to reusable templates.

They also mentioned about the related works and what is lagging in the already existing one as i mentioned in the intro section about the Lida and langchain are the agent which can collaborate only between the single large language model and not with the multiple large language model and however they are very hectic to complete the task and also they talked about the multiagent frameworks like Camel is designed to facilitate the autonomous cooperation among communicative agents through role playing, using inception prompting to guide the chat agents toward task agent and also Oscopilot which is framework for building generalist agents capable of interfacing with elements in the os like web ,code terminal , files, media along with third party applications. The aforementioned are the barriers of the multiagent design process and issues faced by these agents which made them develop the autogen. These are the major, so they take it as the problem statement.

Their main goals is to do rapid prototyping , developer tools for debugging, and reusability.They proposed a solution with the drag and drop feature which allows the developer to make the communication between the multiple large language models and the tools which is already build in within the framework.The tool also consist of the developer debugging to identify the message flow between the large language models and the collaborate tool which helps the developers to find and solve the problem occurring between them in the more quicker manner and also Autogen includes the visualization of the actions by the agents and the metrics like costs, tool invocation and tool output for easy debugging.They also mentioned that it is open source project within the launch of the project over 200k download were happened. They also talked about the core concepts of models, skills, memory, agent, workflow.

Autogen system design consists of the two major parts one is User interface and the second one is the Backend service. The front end consists of the major 4 parts namely, building the workflow between the large language model and the collaborative tools by drag and drop the agents with the parameter tuning and how they have to collaborate with each other can be defined. In the testing and debugging where the developer can visualize and solve the problem between the multi agents workflow. Third, there is a template for the agents which is already built and can be reused. Finally they can deploy the workflow created. In the backend all the model and tools related api keys will be configured and it also supports all the common methods like http, post , get method as well as the crud operation. It also consists of the workflower manager class handles the ingestion of the declarative agents. And the component called the Profiler which handles the messages between the agents and also parses the message to Ui websocket , it will handle the workflow api endpoint.

Finally they talked about the usage and evaluation, in the frameworks have several issues raised by the users of the Autogen that challenges in defining , persisting and reusing component which they resolved with the help of database layer , authoring components resolved by supporting automated tool generators.then component failure is resolved by using test buttons. Based on these issues they integrated the open ai model to solve the issues. They talked the emerging design patterns and research directions including the deployment, collaborative sharing between developers.



Finally they concluded that autogen is capable of solving the problems defined before development and improving the efficiency, faster prototyping ,and easy to debug the issues between the workflow.Additionally, they created this as an Open Source framework because collaborative environments promote faster technological advancements.

References

They referenced the following papers:

1. Harrison Chase. 2022. LangChain. Github. Victor Dibia. 2023. Lida: A tool for automatic generation of grammar-agnostic visualizations and infographics using large language models. arXiv preprint arXiv:2303.02927.
2. Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. arXiv preprint arXiv:2305.14325.
3. Guohao Li, Hasan Hammoud, Hani Itani, DmitriiKhizbullin, and Bernard Ghanem. 2024. Camel: Communicative agents for" mind" exploration of large language model society. Advances in Neural Information Processing Systems, 36.
4. Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang,Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. arXiv preprint arXiv:2305.19118.
5. Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics, 12:157–173.
6. Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. 2023. Augmented language models: a survey. arXiv preprint arXiv:2302.07842.
7. Bo Qiao, Liqun Li, Xu Zhang, Shilin He, Yu Kang,Chaoyun Zhang, Fangkai Yang, Hang Dong, Jue Zhang, Lu Wang, et al. 2023. Taskweaver: A code-first agent framework. arXiv preprint arXiv:2311.17541.
8. 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.