CRESt – Copilot for Real-world Experimental Scientist

Zhichu Ren,[1] Zhen Zhang,[1] Yunsheng Tian[2] and Ju Li[1,3,*]

[1]Department of Materials Science and Engineering, MIT, Cambridge, MA 02139, USA

[2]Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA 02139, USA

[3]Department of Nuclear Science and Engineering, MIT, Cambridge, MA 02139, USA

*Corresponding author: Ju Li, liju@mit.edu

Abstract

Autonomous laboratories[1–5] were previously controlled mainly by scripting languages such as Python, limiting their usage among experimentalists. The recent release of OpenAI's ChatGPT API's function calling feature has enabled seamless integration and execution of Python subroutines in experimental workflows using voice commands.[6] We have developed a system of Copilot for Real-world Experimental Scientist (CRESt) system, with a demonstration shown on YouTube.[7] Large language models (LLMs) empower all research group members, regardless of coding experience, to leverage the robotic platform for their own projects, simply by talking with CRESt.

Methods

The CRESt "operating system" (see YouTube[7]) is mainly composed of four parts: user interface, ChatGPT back-end, active learning and end-effectors. The user interface is based on chatgpt-voice,[8] a platform that enables voice-to-text and text-to-voice interactions. The convenient web framework makes it possible for users to continue the conversation on their cell-phones seamlessly after leaving the physical laboratory, since the back-end is individually hosted and remains unaffected when the front-end is altered. Additionally, local adjustments were implemented to integrate ElevenLabs as the AI voice.[9]

The text message that originates from the front-end is then transmitted to the ChatGPT back-end, which was built on the foundation of CallingGPT.[10] This allows a Python function documented in Google style docstring to be converted into a JSON format recognizable by ChatGPT, which can then be invoked whenever ChatGPT finds it necessary. Furthermore, it closes a feedback loop between ChatGPT and the local Python function, as the suggested function will be immediately executed locally and its return value will be sent back to ChatGPT. One obstacle we faced was that some tasks required the execution of perhaps three functions consecutively. However, ChatGPT didn't always complete all the steps; it might return an incomplete or even failure message after executing one or

two functions. To enhance the robustness of function chain-calling, we implemented a simple trick in CRESt, which is basically adding more “prompt guidance” in the function return message. This approach turned out to be somewhat effective. Specifically, ChatGPT receives one of three templates after it calls a function:

1. Call again: "function successfully called with return value: {ret_val}, please call this function again."
2. Proceed: "function successfully called with return value: {ret_val}, please go to the next step."
3. Function guide: "function successfully called with return value: {ret_val}, please call {function_name} next."

To improve the robustness of the workflow, multiple branches are better to be preset in local Python functions, each ending with one of the templates listed above. For example, if the argument that ChatGPT provides fails to pass the assertion check, it will enter a branch with a “call again” template. With the failure reason included in the return value and “please call this function again” articulated, ChatGPT will be more likely to reattempt the same function, rather than immediately replying to the user with a failure message in text. In the future, it would be highly beneficial if OpenAI could provide more knobs over the function calling API (e.g. probability distribution among functions and the threshold for calling), thereby making the behavior more controllable.

Active learning is considered a good starting point for autonomous experimental science since it works pretty well with small datasets.[11–13] Data acquisition is the most significant challenge in learning projects that involve real-world experiments. Unlike in the virtual world, each datapoint in the real physical world could be fairly expensive and time-consuming to obtain - often a dataset of 1000 points is considered substantial. Given these conditions, the strategy for sampling the design space becomes of paramount importance. The primary function of active learning is to interactively suggest the parameter combination to test in the next batch, backed by rigorous mathematical principles.[14–16] There are various nice frameworks on GitHub, and the one we implemented in CRESt is the Ax platform,[17] developed by a team at Meta, and built upon BoTorch.[18] Ax offers a well-implemented SQL storage option, allowing one to resume the previous active learning campaign even if the GPT backend is reset, by retrieving the history stored in the database.

End-effectors are a set of subroutines ready to be invoked via HTTP requests. Some of these may involve information retrieval tasks (local or public database queries like the Materials Project[19]), while others could have tangible real-world impact, as we have shown in the demo (liquid handling robot, laser cutter, pump, gas valve, robotic arm), primarily the components in data collection. The automation of these devices is mostly handled by PyAutoGUI,[20] a script that can simulate human mouse and keyboard actions. However, we expect this redundant step will eventually become obsolete, as most laboratory equipment should ideally provide a dedicated interface for AI access.

Outlook

What Large Language Models (LLMs) can bring to the realm of science and engineering is a question that we have been pondering since the advent of ChatGPT. There is no question that LLMs has already shown its superb potential as a **literature reviewer**, all one needs is to feed more literature with full content to it. Then, what else? Beyond the role of **experimentalist's assistant**, which we have just developed in the form of CRESt, we envision it will also play a transformative role in at least the following three dimensions:

Instrument coach. Presently, researchers must comprehend the theoretical basis of any technology they wish to utilize, along with the specific operations (sometimes empirical rules or "tricks") of an individual instrument, which can vary significantly from one manufacturer to the other. This latter requirement involves a non-trivial effort, e.g. a series of training sessions for a shared facility or reading a 200-page manual for a group-owned instrument, but is it truly indispensable? We foresee that, in the imminent future, researchers will merely need to articulate their needs in plain language, and LLMs will translate these into the optimal parameter settings (which is what an instrument specialist is doing now). Upon request, the exact part in the manual book can be referred to for the users to further investigate into. Technically, this can be made possible by appropriately fine-tuning an LLM base model by the vendor, which can be done today.

Pipeline diagnostician. LLMs could help identify the root causes of irreproducible results when paired with multi-sensors equipped robots or drones. In the future, an ideal experimentation paradigm is to log the lifetime recording and all the metadata of every sample. When an inexplicable phenomenon occurs, all related logs can be input into the multi-modal LLMs for analysis. Leveraging its superior hypothesis generation ability,[21] the model can propose a list of potential causes, allowing human experts to further investigate the top few that they deem likely. This approach could also be applied in industrial processing pipelines – if a significant drop in the production yield is noticed, LLMs can be employed to identify the "culprit", with human engineers stepping in when complex real-world adjustments are required. This role becomes viable when LLMs can process vast amounts of images (videos), and will be further enhanced when multimodal information (vendor-provided metadata for samples, moisture sensing, sound sensing, etc.) gets well aligned with the visual information.

Mechanism narrator. We anticipate LLMs will excel in applying established scientific principles to novel experimental phenomena. A significant portion of the work in the scientific mechanism exploring stage are pattern matching tasks (e.g. extracting subtle features from a spectrum and comparing with the standard database), which fall within the competence of LLMs. A typical procedure could be as straightforward as asking LLMs: *a sample with this composition and processing history exhibited superior performance, here are all the characterization results (Scanning Electron Microscope, X-Ray Diffraction, etc.), please explain why this result is so good.* Human researchers may examine the most reasonable explanations from a range of narratives LLMs generate, and start the scientific

discussions from there. However, this process will pose the greatest challenge, with requirements including (i) image input and alignment with scientific terms, (ii) information retrieval capabilities from online physical scientific databases, (iii) LLMs being pretrained on scientific journal main context and supplements, (iv) cutting-edge subfield ML models invokable as plug-ins.

CRESt is just a starting point of how LLMs can assist experimental scientists, and we believe the true potential of LLMs lies in their hypothesis-generating capability.[21] Humans possess a relatively limited knowledge base but exceptional causal inference capabilities, allowing us to produce pinpoint hypotheses, albeit not in large quantities. In contrast, relying on the extensive knowledge base and the emerging ability to discover the key pattern in a large datasheet via the Excel plugin,[22] AI can generate numerous hypotheses in little time, but is usually less discerning today.[21] Therefore, this is not a story of AI competing with humans, but rather one of AI complementing humans. In the paradigm of "**AI suggests, humans select**", we hope the strength of both parties can be leveraged.

Acknowledgments

We thank Ali Abdelhafiz, Wenhao Gao, Zhewen Guo, Chiawei Hsu, Russell Scott In, Hang Jiang, Yichen Li, Ji Lin, David Liu, Yuyang Liu, Elton Pan, Yangjeong Park, Zekun Ren, Zhaofeng Wu, Hongbin Xu, Jialiang Zhao, Daniel Zheng for insightful discussions. We acknowledge support by DTRA (Award No. HDTRA1-20-2-0002) Interaction of Ionizing Radiation with Matter (IIRM) University Research Alliance (URA).

References

1. Coley, C. W. *et al.* A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science* 365, eaax1566 (2019).

2. Sun, S. *et al.* Accelerated Development of Perovskite-Inspired Materials via High-Throughput Synthesis and Machine-Learning Diagnosis. *Joule* 3, 1437–1451 (2019).

3. Burger, B. *et al.* A mobile robotic chemist. *Nature* 583, 237–241 (2020).

4. MacLeod, B. P. *et al.* Self-driving laboratory for accelerated discovery of thin-film materials. *Sci Adv* 6, eaaz8867 (2020).

5. Ceder, G. AI-driven robots start hunting for novel materials without help from humans. *Science* (2023) doi:10.1126/science.adi3613.

6. Function calling and other API updates. https://openai.com/blog/function-calling-and-other-api-updates (2023).

7. Ren, Z. CRESt - Copilot for Real-world Experimental Scientist. https://youtu.be/POPPVtGueb0 (2023).

8. Nguyen, S. chatgpt-voice. https://chatgpt.sonng.dev/ (2023).

9. Eleven Labs. https://beta.elevenlabs.io/.

10. Qin, J. CallingGPT. https://github.com/RockChinQ/CallingGPT (2023).

11. Stach, E. *et al.* Autonomous experimentation systems for materials development: A community perspective. *Matter* 4, 2702–2726 (2021).

12. Coley, C. W., Eyke, N. S. & Jensen, K. F. Autonomous Discovery in the Chemical Sciences Part I: Progress. *Angewandte Chemie Int Ed* 59, 22858–22893 (2020).

13. Stein, H. S. & Gregoire, J. M. Progress and prospects for accelerating materials science with automated and autonomous workflows. *Chem Sci* 10, 9640–9649 (2019).

14. Frazier, P. I. A Tutorial on Bayesian Optimization. *Arxiv* (2018) doi:10.48550/arxiv.1807.02811.

15. Williams, C. E. R. & C. K. I. *Gaussian Processes for Machine Learning*. (1996).

16. Morgan, D. *et al.* Machine learning in nuclear materials research. *Curr Opin Solid State Mater Sci* 26, 100975 (2022).

17. Adaptive Experimentation Platform. https://ax.dev/.

18. Balandat, M. *et al.* BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. *Arxiv* (2019) doi:10.48550/arxiv.1910.06403.

19. Jain, A. *et al.* Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *Apl Mater* 1, 011002 (2013).

20. Sweigart, A. PyAutoGUI. https://github.com/asweigart/pyautogui.

21. Park, Y. J. *et al.* Can ChatGPT be used to generate scientific hypotheses? *arXiv* (2023) doi:10.48550/arxiv.2304.12208.

22. Brockman, G. The Inside Story of ChatGPT's Astonishing Potential. https://www.youtube.com/watch?v=C_78DM8fG6E&;ab_channel=TED (2023).