**Materials Agent: An LLM-Based Agent with Tool-Calling Capabilities for Cheminformatics**

Out-of-the box large language model (LLM) implementations such as ChatGPT, while offering interesting responses, generally provide little to no control over the workflow of the LLM by which the response is generated. In other words, it is easy to get LLMs to say something in response to a prompt, but difficult to get them to do something via an expected workflow. A solution to do this problem is to equip LLMs with tool-calling capabilities; i.e., allow the LLM to generate the response via an available function(s) which is (are) appropriate to answer the prompt. An LLM with tool-calling capabilities, when prompted, typically decides which tool(s) to call (execute) and the order in which to execute them, along with the arguments to pass to them. It then executes these and returns the response. Such a system offers several powerful capabilities such as the ability to query databases to return the latest information and reliably perform mathematical calculations. Such capabilities have been incorporated into ChatGPT via plugins such as Expedia and Wolfram, among others.[Ref: <https://openai.com/index/chatgpt-plugins/>] In the chemistry literature, recent attempts have been made by researchers such as Smit and coworkers, Schwaller and coworkers, among others.[Ref.: <https://www.nature.com/articles/s42256-023-00788-1>, https://www.nature.com/articles/s42256-024-00832-8]

Through Materials Agent, which is an LLM-based agent with tool-calling capabilities for cheminformatics, we seek to build on these attempts to provide a variety of important tool-calling capabilities and build a framework to expand on these. We hope that this can serve to increase LLM adoption in the community and lower the barrier to entry for cheminformatics. Materials Agent is built using the LangChain library [Ref. https://www.langchain.com/], GPT 3.5-turbo[Ref.: https://platform.openai.com/docs/models] as the underlying LLM, and user interface is based on the FastDash project [Ref.: https://docs.fastdash.app/]. In this demonstration, we have provided tools based on RDKit[TODO: RDKit: Open-source cheminformatics; http://www.rdkit.org]—a popular cheminformatics library, some custom tools, as well as a Retrieval Augmented Generation (RAG) to allow LLM to interact with documents. The full code is available at <https://github.com/dkedar7/materials-agent> and the working application, hosted on Google Cloud Platform (GCP), is available at <https://materials-agent-hpn4y2dvda-ue.a.run.app/>. The demonstration video is uploaded on YouTube at <https://www.youtube.com/watch?v=_5yCOg5Bi_Q&ab_channel=ArchitDatar>. In the following section, we describe some key use cases.

Equipping LLMs with tools based on standard cheminformatics packages (RDKit)

Common cheminformatics workflows involve obtaining SMILES strings and key properties of molecules. Searching for these data individually can be time consuming. The RDKit *CalcMolDescriptors* module comes pre-loaded with these data based on SMILES strings. We created a function to select 19 of the most common properties (for ease of visualization) and added it as a tool to the LLM. Not only can the LLM perform the simple task of providing these descriptions when a SMILES string is input, but it can also perform more complicated tasks involving this functionality. As shown in Figure 1 below, given a complex query, the LLM breaks it down into individual tasks, utilizes these tools in the correct sequence, and renders a response.



Figure 1. Workflow of a response generated to a user prompt by Materials Agent using a tool based on RDKit.

Tool-calling for custom tools: Radial Distribution Function calculation

Combining custom tools with LLMs lead to some interesting advantages. For one, they offer the makers of these custom tools an ability to provide their users with a more intuitive natural language-based user interface which can lower the barrier to entry and increase adoption. Other benefits can be that they can be integrated into other workflows and automate more work. To demonstrate this, we have built a tool which computes and plots the radial distribution function (RDF) and integrated it with an LLM. The RDF is an important function commonly used in molecular dynamics and Monte Carlo simulations to understand the statistics of distances between two particles averaged over the simulation. By studying these, one can understand the nature of interactions between these particles. Here, we constructed a custom tool to compute RDF for the distance between a water molecule and the framework atoms in a metal-organic framework (MOF) in a single molecule canonical ensemble Monte Carlo simulation. The inputs are a PDB file of the MOF structure and a TXT file containing snapshots of the location of the water molecule during the simulation. The distance computation also accounts for triclinic periodic boundary conditions which is the accurate way to quantify distances for crystalline systems such as this one. The inputs and outputs for this tool are shown below in Figure 2(a).

Furthermore, we also stress that this approach is easily scalable and transferable, and adding new tools is exceedingly easy. The reader is encouraged to clone our GitHub repository and experiment with adding new tools to this software package. New tools can be easily added to the *src/tools.py* file in the repository via the format shown in the code snippet in Figure 2(b).



Figure 2. Niche purpose tools with LLM. (a) Illustration of the RDF computing tool. (b) Code snippet to highlight the ease of transferability of LLMs with tool-calling capabilities.

RAG capabilities

Summarizing and asking questions of documents is another common LLM use case. This capability is provided out-of-the box through the EmbedChain library [Ref: Singh, Taranjeet Embedchain, <https://github.com/embedchain/embedchain>.] and we have integrated that into Materials Agent for convenience. We demonstrate the utility of this by supplying the LLM with a URL to a materials and safety datasheet (MSDS) and asking questions of it (see Fig 3).



Figure 3. Illustration of RAG for interacting with a MSDS.

In future, we aim to expand the toolkit that the LLM is equipped with. For instance, we can add functions built on the publicly available PubChem database.[Ref.: <https://pubchem.ncbi.nlm.nih.gov/docs/programmatic-access>] as well as some functions built off of it. [Reference. <https://pubchempy.readthedocs.io/en/latest/guide/introduction.html>] We also aim to train it on user manuals of commonly used molecular simulations software such as GROMACS[Ref.: https://doi.org/10.5281/zenodo.11148638.], RASPA[Ref.: https://www.tandfonline.com/doi/full/10.1080/08927022.2015.1010082], and QuantumEspresso[Ref.: https://iopscience.iop.org/article/10.1088/0953-8984/21/39/395502] to assist with setting up molecular simulations.

Through the experience of building Materials Agent, we realized that, while convenient, such agents cannot replace the need for human vigilance. At the same time, having the development of such an agent will make cheminformatics utilities easier to access for a broader range of users, lower the barrier to entry, and ultimately, accelerate the pace of materials development.