Thoughts/ideas for function calling sync with Gorilla folks:

- Can we fine-tune a language model that can convert API urls into a JSON format suitable to train an LLM on ? This will enable augmentation of existing API/function datasets on a large scale + very fast.
- There are many existing open/free API datasets - API-bank, ToolAlpaca, etc. Why didn’t the Gorilla folks choose to augment their API-bench data with these? Anything to do with the distribution or quality of data? Any issues they ran into while trying to train on these particular datasets? Or is it just because they were not aware/ didn’t have the time?
- How are the Gorilla folks working on sequence of API calls? Is it just independent calls of multiple APIs ? Or is it actually sequential ? If the former, why - what is the use case of that ? If the latter, what exactly is their approach and how did they build the dataset for it?

(**Unrelated to Gorilla**) Long term vision - try to build a representation space for APIs which will enable large datasets to be built by encoding several APIs based on their documentation and use-case. High-dimensional latent space representations further allow powerful ML models to be trained on them leading to more capabilities