Gorilla : Connecting LLM with Massive APIs

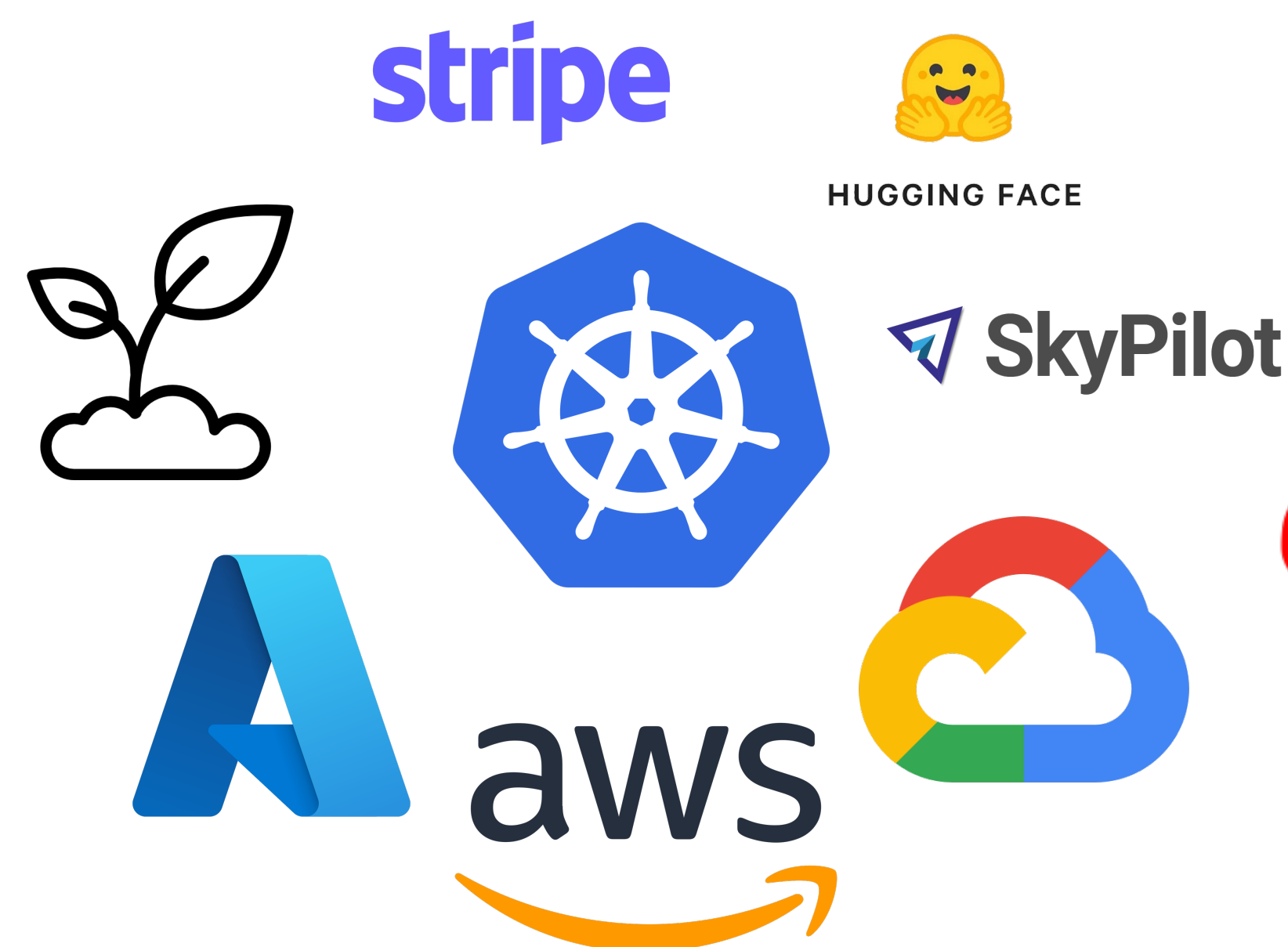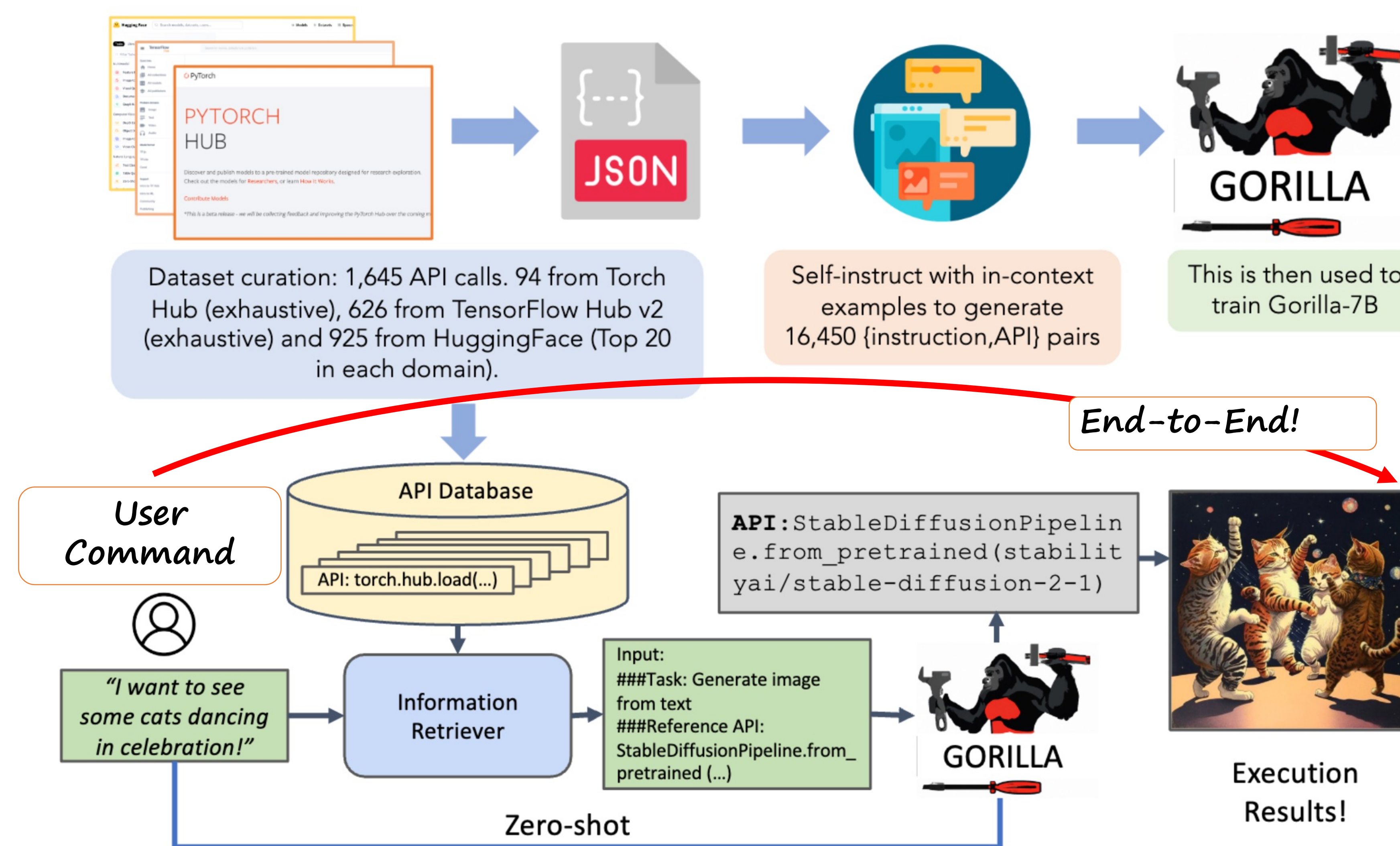🌐 gorilla.cs.berkeley.edu   ✉ sgp@berkeley.edu

GORILLA

Overview

In order to use **Tools**, people currently have to read **long documentations** to understand their usages. How can **LLM** help us resolve this, improve our productivity?
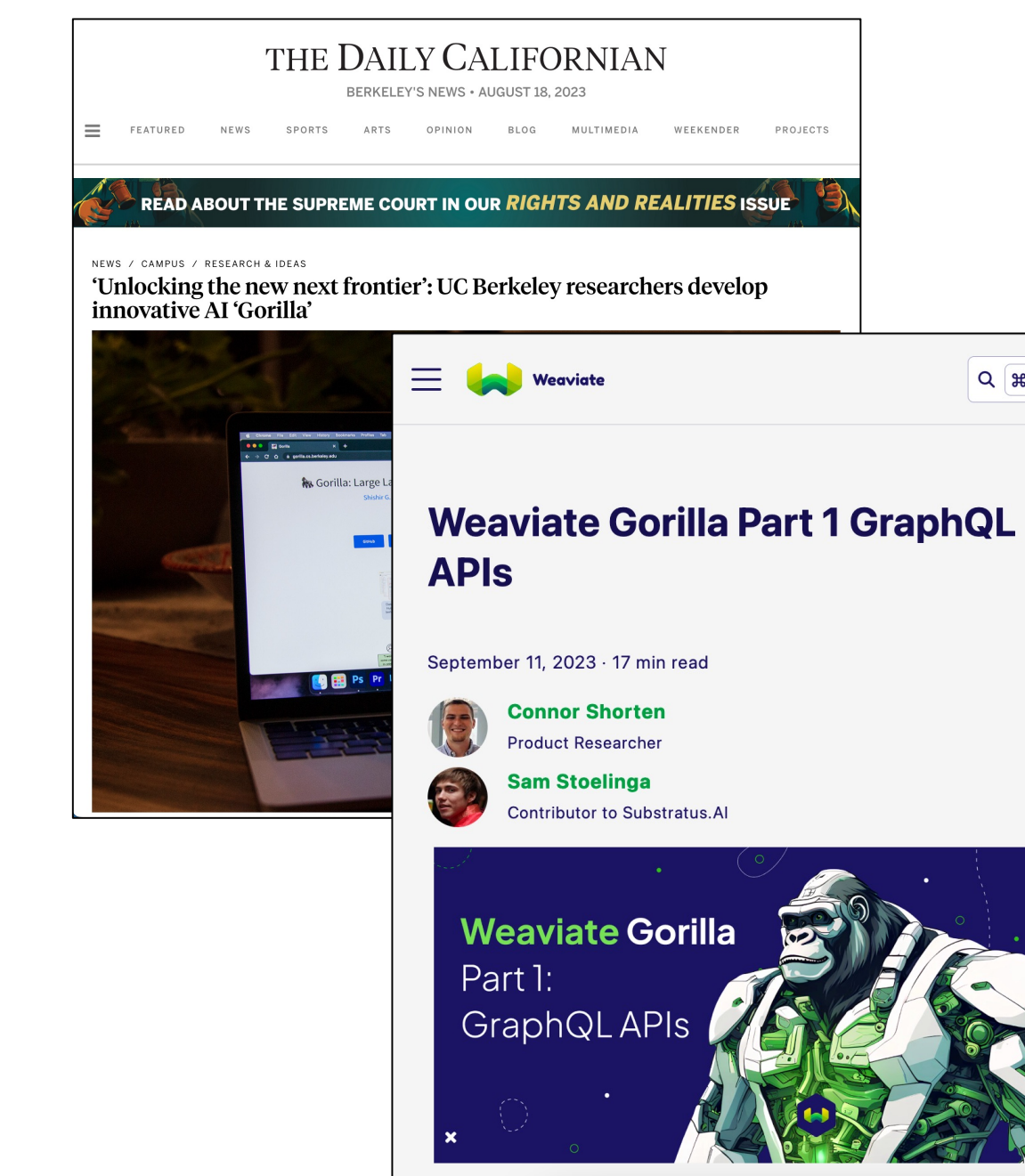

Which to use and how to use them?

Gorilla Workflow – Self-Instruction + Finetuning

Dataset curation: 1,645 API calls. 94 from Torch Hub (exhaustive), 626 from TensorFlow Hub v2 (exhaustive) and 925 from HuggingFace (Top 20 in each domain).

Self-instruct with in-context examples to generate 16,450 {instruction,API} pairs

This is then used to train Gorilla-7B

End-to-End!

User Command

API Database
API: torch.hub.load(...)

"I want to see some cats dancing in celebration!"

Information Retriever

Input:
###Task: Generate image from text
###Reference API: StableDiffusionPipeline.from_pretrained (...)

API: StableDiffusionPipeline.from_pretrained(stability ai/stable-diffusion-2-1)
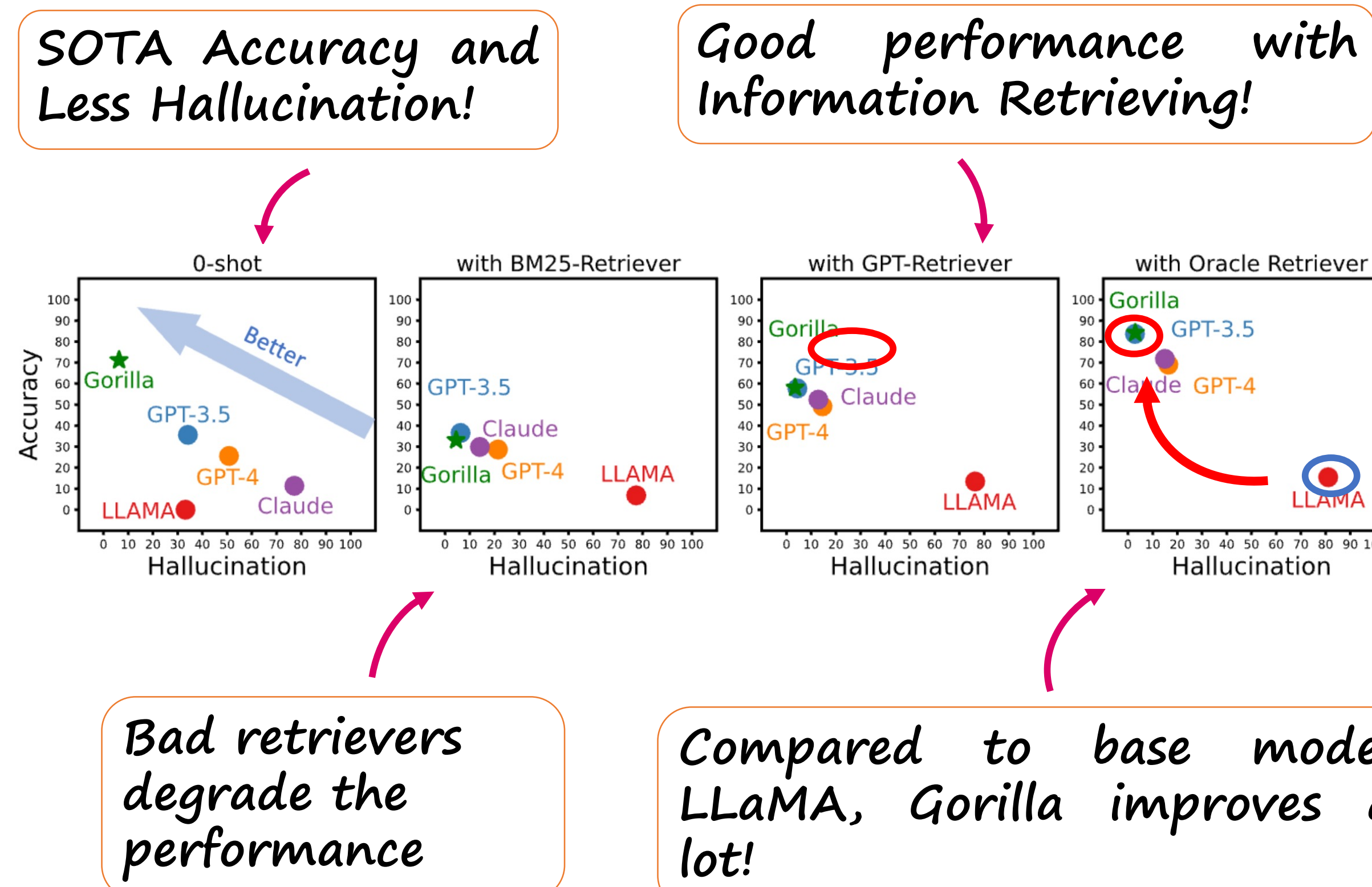
Execution Results!

Zero-shot

Impact

Used at: Adobe, Apple, CMU, CISCO, Cloudera, Dropbox, UCLA, GNU, IBM, Intel, Intuit, LinkedIn, Microsoft, MIT, Netflix, Nvidia, Salesforce, Tesla, …

Gorilla: LLM + APIs

✕ Reading long documentations to understand how to use tools (e.g., API calls) is very hard

⚠ **Current LLMs (e.g., GPT-4) cannot reliably generate API calls**

💡 **Gorilla** is a finetuned LLaMA model using a collected API dataset, solving the problem with a **Natural Language to API Call** interface

✓ **Gorilla outperforms GPT-4** on various API benchmarks we constructed!

How does Gorilla Perform?

SOTA Accuracy and Less Hallucination!

Good performance with Information Retrieving!

Bad retrievers degrade the performance

Compared to base model LLaMA, Gorilla improves a lot!

Does Gorilla adapt to evolving APIs?

| | Default Response | Update the model | Update the model repository |

USER: I want to automatically remove the background from an input image. <Retrieval: ..fcn_resnet50..>

GORILLA:
api_call:
    "torch.hub.load('pytorch/vision', 'fcn_resnet50', pretrained=True)"

USER: I want to automatically remove the background from an input image. <Retrieval: ..fcn_resnet101..>

GORILLA:
api_call:
    "torch.hub.load('pytorch/vision', 'fcn_resnet101', pretrained=True)"

USER: I want to automatically remove the background from an input image. <Retrieval: ..NVIDIA/DeepLearningExamples:torchhub..>

GORILLA:
api_call:
    "torch.hub.load('NVIDIA/DeepLearningExamples:torchhub', 'fcn_resnet50', pretrained=True)"

Gorilla can adapt to test time document changes without any further finetuning. This implies developers/users can flexibly update their API usage document and Gorilla is still being able to use it.