🏠   Modules      Agents      Agent types      Structured tool chat

# Structured tool chat

The structured tool chat agent is capable of using multi-input tools.

Older agents are configured to specify an action input as a single string, but this agent can use the provided tools' `args_schema` to populate the action input.

This functionality is natively available using agent types: `structured-chat-zero-shot-react-description` or `AgentType.STRUCTURED_CHAT_ZERO_SHOT_REACT_DESCRIPTION`

```python
import os
os.environ["LANGCHAIN_TRACING"] = "true" # If you want to trace the
execution of the program, set to "true"
```

```python
from langchain.agents import AgentType
from langchain.chat_models import ChatOpenAI
from langchain.agents import initialize_agent
```

## Initialize Tools

We will test the agent using a web browser.

```python
from langchain.agents.agent_toolkits import PlayWrightBrowserToolkit
from langchain.tools.playwright.utils import (
    create_async_playwright_browser,
    create_sync_playwright_browser, # A synchronous browser is available,
though it isn't compatible with jupyter.
)

# This import is required only for jupyter notebooks, since they have
their own eventloop
import nest_asyncio
nest_asyncio.apply()
```

```python
async_browser = create_async_playwright_browser()
browser_toolkit =
PlayWrightBrowserToolkit.from_browser(async_browser=async_browser)
tools = browser_toolkit.get_tools()
```

```python
llm = ChatOpenAI(temperature=0) # Also works well with Anthropic models
agent_chain = initialize_agent(tools, llm,
agent=AgentType.STRUCTURED_CHAT_ZERO_SHOT_REACT_DESCRIPTION, verbose=True)
```

```python
response = await agent_chain.arun(input="Hi I'm Erica.")
print(response)
```

```
    > Entering new AgentExecutor chain...
    Action:
    ```
    {
      "action": "Final Answer",
      "action_input": "Hello Erica, how can I assist you today?"
    }
    ```


    > Finished chain.
    Hello Erica, how can I assist you today?
```

```python
response = await agent_chain.arun(input="Don't need help really just
chatting.")
print(response)
```

```
    > Entering new AgentExecutor chain...
```

```
    > Finished chain.
    I'm here to chat! How's your day going?
```

```python
response = await agent_chain.arun(input="Browse to blog.langchain.dev and
summarize the text, please.")
print(response)
```

```
    > Entering new AgentExecutor chain...
    Action:
    ```
    {
      "action": "navigate_browser",
      "action_input": {
        "url": "https://blog.langchain.dev/"
      }
    }
    ```


    Observation: Navigating to https://blog.langchain.dev/ returned status
code 200
    Thought:I need to extract the text from the webpage to summarize it.
    Action:
    ```
    {
      "action": "extract_text",
      "action_input": {}
    }
    ```


    Observation: LangChain LangChain Home About GitHub Docs LangChain The
official LangChain blog. Auto-Evaluator Opportunities Editor's Note: this
is a guest blog post by Lance Martin.


    TL;DR


    We recently open-sourced an auto-evaluator tool for grading LLM
question-answer chains. We are now releasing an open source, free to use
hosted app and API to expand usability. Below we discuss a few
```

opportunities to further improve May 1, 2023 5 min read Callbacks Improvements TL;DR: We're announcing improvements to our callbacks system, which powers logging, tracing, streaming output, and some awesome third-party integrations. This will better support concurrent runs with independent callbacks, tracing of deeply nested trees of LangChain components, and callback handlers scoped to a single request (which is super useful for May 1, 2023 3 min read Unleashing the power of AI Collaboration with Parallelized LLM Agent Actor Trees Editor's note: the following is a guest blog post from Cyrus at Shaman AI. We use guest blog posts to highlight interesting and novel applications, and this is certainly that. There's been a lot of talk about agents recently, but most have been discussions around a single agent. If multiple Apr 28, 2023 4 min read Gradio & LLM Agents Editor's note: this is a guest blog post from Freddy Boulton, a software engineer at Gradio. We're excited to share this post because it brings a large number of exciting new tools into the ecosystem. Agents are largely defined by the tools they have, so to be able to equip Apr 23, 2023 4 min read RecAlign — The smart content filter for social media feed [Editor's Note] This is a guest post by Tian Jin. We are highlighting this application as we think it is a novel use case. Specifically, we think recommendation systems are incredibly impactful in our everyday lives and there has not been a ton of discourse on how LLMs will impact Apr 22, 2023 3 min read Improving Document Retrieval with Contextual Compression Note: This post assumes some familiarity with LangChain and is moderately technical.

💡 TL;DR: We've introduced a new abstraction and a new document Retriever to facilitate the post-processing of retrieved documents. Specifically, the new abstraction makes it easy to take a set of retrieved documents and extract from them Apr 20, 2023 3 min read Autonomous Agents & Agent Simulations Over the past two weeks, there has been a massive increase in using LLMs in an agentic manner. Specifically, projects like AutoGPT, BabyAGI, CAMEL, and Generative Agents have popped up. The LangChain community has now implemented some parts of all of those projects in the LangChain framework. While researching and Apr 18, 2023 7 min read AI-Powered Medical Knowledge: Revolutionizing Care for Rare Conditions [Editor's Note]: This is a guest post by Jack Simon, who recently participated in a hackathon at Williams College. He built a LangChain-powered chatbot focused on appendiceal cancer, aiming to make specialized knowledge more accessible to those in need. If you are interested in building a chatbot for another rare Apr 17, 2023 3 min read Auto-Eval of Question-Answering Tasks By Lance Martin

Context

LLM ops platforms, such as LangChain, make it easy to assemble LLM components (e.g., models, document retrievers, data loaders) into chains. Question-Answering is one of the most popular applications of these chains. But it is often not always obvious to determine what parameters (e.g. Apr 15, 2023 3 min read Announcing LangChainJS Support for Multiple JS Environments TLDR: We're announcing support for running LangChain.js in browsers, Cloudflare Workers, Vercel/Next.js, Deno, Supabase Edge Functions, alongside existing support for Node.js ESM and CJS. See install/upgrade docs and breaking changes list.

Context

Originally we designed LangChain.js to run in Node.js, which is the Apr 11, 2023 3 min read LangChain x Supabase Supabase is holding an AI Hackathon this week. Here at LangChain we are big fans of both Supabase and hackathons, so we thought this would be a perfect time to highlight the multiple ways you can use LangChain and Supabase together.

The reason we like Supabase so much is that Apr 8, 2023 2 min read Announcing our $10M seed round led by Benchmark It was only six months ago that we released the first version of LangChain, but it seems like several years. When we launched, generative AI was starting to go mainstream: stable diffusion had just been released and was captivating people's imagination and fueling an explosion in developer activity, Jasper Apr 4, 2023 4 min read Custom Agents One of the most common requests we've heard is better functionality and documentation for creating custom agents. This has always been a bit tricky — because in our mind it's actually still very unclear what an "agent" actually is, and therefore what the "right" abstractions for them may be. Recently, Apr 3, 2023 3 min read Retrieval TL;DR: We are adjusting our abstractions to make it easy for other retrieval methods besides the LangChain VectorDB object to be used in LangChain. This is done with the goals of (1) allowing retrievers constructed elsewhere to be used more easily in LangChain, (2) encouraging more experimentation with alternative Mar 23, 2023 4 min read LangChain + Zapier Natural Language Actions (NLA) We are super excited to team up with Zapier and integrate their new Zapier NLA API into LangChain, which you can now use with your agents and chains. With this integration, you have access to the 5k+ apps and 20k+ actions on Zapier's platform through a natural language API interface. Mar 16, 2023 2 min read Evaluation Evaluation of language models, and by extension applications built on top of language models, is hard. With recent model releases (OpenAI, Anthropic, Google) evaluation is becoming a bigger and bigger issue. People are starting to try to tackle this, with OpenAI releasing

OpenAI/evals — focused on evaluating OpenAI models. Mar 14, 2023 3 min read LLMs and SQL Francisco Ingham and Jon Luo are two of the community members leading the change on the SQL integrations. We're really excited to write this blog post with them going over all the tips and tricks they've learned doing so. We're even more excited to announce that we' Mar 13, 2023 8 min read Origin Web Browser [Editor's Note]: This is the second of hopefully many guest posts. We intend to highlight novel applications building on top of LangChain. If you are interested in working with us on such a post, please reach out to harrison@langchain.dev.

Authors: Parth Asawa (pgasawa@), Ayushi Batwara (ayushi.batwara@), Jason Mar 8, 2023 4 min read Prompt Selectors One common complaint we've heard is that the default prompt templates do not work equally well for all models. This became especially pronounced this past week when OpenAI released a ChatGPT API. This new API had a completely new interface (which required new abstractions) and as a result many users Mar 8, 2023 2 min read Chat Models Last week OpenAI released a ChatGPT endpoint. It came marketed with several big improvements, most notably being 10x cheaper and a lot faster. But it also came with a completely new API endpoint. We were able to quickly write a wrapper for this endpoint to let users use it like Mar 6, 2023 6 min read Using the ChatGPT API to evaluate the ChatGPT API OpenAI released a new ChatGPT API yesterday. Lots of people were excited to try it. But how does it actually compare to the existing API? It will take some time before there is a definitive answer, but here are some initial thoughts. Because I'm lazy, I also enrolled the help Mar 2, 2023 5 min read Agent Toolkits Today, we're announcing agent toolkits, a new abstraction that allows developers to create agents designed for a particular use-case (for example, interacting with a relational database or interacting with an OpenAPI spec). We hope to continue developing different toolkits that can enable agents to do amazing feats. Toolkits are supported Mar 1, 2023 3 min read TypeScript Support It's finally here... TypeScript support for LangChain.

What does this mean? It means that all your favorite prompts, chains, and agents are all recreatable in TypeScript natively. Both the Python version and TypeScript version utilize the same serializable format, meaning that artifacts can seamlessly be shared between languages. As an Feb 17, 2023 2 min read Streaming Support in LangChain We're excited to announce streaming support in LangChain. There's been a lot of talk about the best UX for LLM applications, and we believe streaming is at its core. We've also updated the chat-langchain repo to include streaming and async execution. We hope that this repo can serve Feb 14, 2023 2 min read LangChain + Chroma Today we're announcing LangChain's integration with Chroma, the first step on the path to the Modern A.I Stack.

LangChain — The A.I—native developer toolkit

We started LangChain with the intent to build a modular and flexible framework for developing A.I—native applications. Some of the use cases Feb 13, 2023 2 min read Page 1 of 2 Older Posts → LangChain © 2023 Sign up Powered by Ghost
    Thought:
    > Finished chain.
    The LangChain blog has recently released an open—source auto—evaluator tool for grading LLM question—answer chains and is now releasing an open—source, free—to—use hosted app and API to expand usability. The blog also discusses various opportunities to further improve the LangChain platform.

```python
response = await agent_chain.arun(input="What's the latest xkcd comic about?")
print(response)
```

```
    > Entering new AgentExecutor chain...
    Thought: I can navigate to the xkcd website and extract the latest comic title and alt text to answer the question.
    Action:
    ```
    {
      "action": "navigate_browser",
      "action_input": {
        "url": "https://xkcd.com/"
      }
    }
    ```

    Observation: Navigating to https://xkcd.com/ returned status code 200
    Thought:I can extract the latest comic title and alt text using CSS selectors.
    Action:
    ```
    {
      "action": "get_elements",
      "action_input": {
```

```
      "selector": "#ctitle, #comic img",
      "attributes": ["alt", "src"]
    }
  }
  ```

  Observation: [{"alt": "Tapetum Lucidum", "src":
"//imgs.xkcd.com/comics/tapetum_lucidum.png"}]
  Thought:
  > Finished chain.
  The latest xkcd comic is titled "Tapetum Lucidum" and the image can be
found at https://xkcd.com/2565/.
```

# Adding in memory

Here is how you add in memory to this agent

```python
from langchain.prompts import MessagesPlaceholder
from langchain.memory import ConversationBufferMemory
```

```python
chat_history = MessagesPlaceholder(variable_name="chat_history")
memory = ConversationBufferMemory(memory_key="chat_history",
return_messages=True)
```

```python
agent_chain = initialize_agent(
    tools,
    llm,
    agent=AgentType.STRUCTURED_CHAT_ZERO_SHOT_REACT_DESCRIPTION,
    verbose=True,
    memory=memory,
    agent_kwargs = {
        "memory_prompts": [chat_history],
        "input_variables": ["input", "agent_scratchpad", "chat_history"]
    }
)
```

```python
response = await agent_chain.arun(input="Hi I'm Erica.")
print(response)
```

```
    > Entering new AgentExecutor chain...
    Action:
    ```
    {
      "action": "Final Answer",
      "action_input": "Hi Erica! How can I assist you today?"
    }
    ```


    > Finished chain.
    Hi Erica! How can I assist you today?
```

```python
response = await agent_chain.arun(input="whats my name?")
print(response)
```

```
    > Entering new AgentExecutor chain...
    Your name is Erica.

    > Finished chain.
    Your name is Erica.
```