🏠 Modules Agents Agent Types

JSON Chat Agent

JSON Chat Agent

Some language models are particularly good at writing JSON. This agent uses JSON to format its outputs, and is aimed at supporting Chat Models.

```
from langchain import hub
from langchain.agents import AgentExecutor, create_json_chat_agent
from langchain_community.tools.tavily_search import TavilySearchResults
from langchain_openai import ChatOpenAI
```

Initialize Tools

We will initialize the tools we want to use

```
tools = [TavilySearchResults(max_results=1)]
```

Create Agent

```
# Get the prompt to use - you can modify this!
prompt = hub.pull("hwchase17/react-chat-json")
```

```
# Choose the LLM that will drive the agent
llm = ChatOpenAI()

# Construct the JSON agent
agent = create_json_chat_agent(llm, tools, prompt)
```

Run Agent

```
# Create an agent executor by passing in the agent and tools
agent_executor = AgentExecutor(
    agent=agent, tools=tools, verbose=True, handle_parsing_errors=True
)
```

```
agent_executor.invoke({"input": "what is LangChain?"})
```

```
> Entering new AgentExecutor chain...
{
    "action": "tavily_search_results_json",
    "action_input": "LangChain"
}[{'url':
'https://www.ibm.com/topics/langchain',
'content': 'LangChain is essentially a
library of abstractions for Python and
Javascript, representing common steps and
concepts  LangChain is an open source
orchestration framework for the development
of applications using large language models
other LangChain features, like the eponymous
chains.  LangChain provides integrations for
over 25 different embedding methods, as well
as for over 50 different vector
storesLangChain is a tool for building
applications using large language models
(LLMs) like chatbots and virtual agents. It
simplifies the process of programming and
integration with external data sources and
software workflows. It supports Python and
Javascript languages and supports various LLM
providers, including OpenAI, Google, and
IBM.'}]{
    "action": "Final Answer",
    "action_input": "LangChain is an open
source orchestration framework for the
development of applications using large
language models. It simplifies the process of
programming and integration with external
data sources and software workflows. It
supports Python and Javascript languages and
supports various LLM providers, including
OpenAI, Google, and IBM."
}

> Finished chain.
```

```
{'input': 'what is LangChain?',
 'output': 'LangChain is an open source
orchestration framework for the development
of applications using large language models.
It simplifies the process of programming and
integration with external data sources and
software workflows. It supports Python and
Javascript languages and supports various LLM
providers, including OpenAI, Google, and
IBM.'}
```

Using with chat history

```
from langchain_core.messages import
AIMessage, HumanMessage
```

```
agent_executor.invoke(
    {
        "input": "what's my name?",
        "chat_history": [
            HumanMessage(content="hi! my name is bob"),
            AIMessage(content="Hello Bob! How can I assist you today?"),
        ],
    }
)
```

```


> Entering new AgentExecutor chain...
Could not parse LLM output: It seems that you have already mentioned your name as Bob. Therefore, your name is Bob. Is there anything else I can assist you with?Invalid or incomplete response{
    "action": "Final Answer",
    "action_input": "Your name is Bob."
}

> Finished chain.
```

```
{'input': "what's my name?",
 'chat_history': [HumanMessage(content='hi!
my name is bob'),
  AIMessage(content='Hello Bob! How can I
assist you today?')],
 'output': 'Your name is Bob.'}
```