Modules Agents Agent Types ReAct

ReAct

This walkthrough showcases using an agent to implement the ReAct logic.

```
from langchain import hub
from langchain.agents import AgentExecutor, create_react_agent
from langchain_community.tools.tavily_search import TavilySearchResults
from langchain_openai import OpenAI
```

Initialize tools

Let’s load some tools to use.

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

Create Agent

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

```
# Choose the LLM to use
llm = OpenAI()

# Construct the ReAct agent
agent = create_react_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)
```

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

```
> Entering new AgentExecutor chain...
 I should research LangChain to learn more
about it.
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.'}] I should read the summary and look at
the different features and integrations of
LangChain.
Action: tavily_search_results_json
Action Input: "LangChain features and
integrations"[{'url':
'https://www.ibm.com/topics/langchain',
'content': "LangChain provides integrations
for over 25 different embedding methods, as
well as for over 50 different vector stores
LangChain is an open source orchestration
framework for the development of applications
using large language models  other LangChain
features, like the eponymous chains.
LangChain is essentially a library of
abstractions for Python and Javascript,
representing common steps and
conceptsLaunched by Harrison Chase in October
2022, LangChain enjoyed a meteoric rise to
prominence: as of June 2023, it was the
single fastest-growing open source project on
Github. 1 Coinciding with the momentous
launch of OpenAI's ChatGPT the following
month, LangChain has played a significant
role in making generative AI more accessible
to enthusias..."}] I should take note of the
launch date and popularity of LangChain.
Action: tavily_search_results_json
Action Input: "LangChain launch date and
popularity"[{'url':
'https://www.ibm.com/topics/langchain',
'content': "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 stores
```

```
LangChain is essentially a library of
abstractions for Python and Javascript,
representing common steps and
conceptsLaunched by Harrison Chase in October
2022, LangChain enjoyed a meteoric rise to
prominence: as of June 2023, it was the
single fastest-growing open source project on
Github. 1 Coinciding with the momentous
launch of OpenAI's ChatGPT the following
month, LangChain has played a significant
role in making generative AI more accessible
to enthusias..."}] I now know the final
answer.
Final Answer: LangChain is an open source
orchestration framework for building
applications using large language models
(LLMs) like chatbots and virtual agents. It
was launched by Harrison Chase in October
2022 and has gained popularity as the
fastest-growing open source project on Github
in June 2023.

> Finished chain.
```

```
{'input': 'what is LangChain?',
 'output': 'LangChain is an open source
orchestration framework for building
applications using large language models
(LLMs) like chatbots and virtual agents. It
was launched by Harrison Chase in October
2022 and has gained popularity as the
fastest-growing open source project on Github
in June 2023.'}
```

Using with chat history

When using with chat history, we will need a prompt that takes that into account

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

```
# Construct the ReAct agent
agent = create_react_agent(llm, tools,
prompt)
agent_executor = AgentExecutor(agent=agent,
tools=tools, verbose=True)
```

```
from langchain_core.messages import
AIMessage, HumanMessage

agent_executor.invoke(
    {
        "input": "what's my name? Only use a
tool if needed, otherwise respond with Final Answer",
        # Notice that chat_history is a string, since this prompt is aimed at LLMs, not chat models
        "chat_history": "Human: Hi! My name is Bob\nAI: Hello Bob! Nice to meet you",
    }
)
```

```



> Entering new AgentExecutor chain...
Thought: Do I need to use a tool? No
Final Answer: Your name is Bob.

> Finished chain.
```

```
{'input': "what's my name? Only use a tool if needed, otherwise respond with Final Answer",
 'chat_history': 'Human: Hi! My name is Bob\nAI: Hello Bob! Nice to meet you',
 'output': 'Your name is Bob.'}
```