

react-hierarchy-CsPbI3-bandgap

November 29, 2023

```
[ ]: import os

from dotenv import load_dotenv
from langchain import hub
from langchain.agents import AgentExecutor, AgentType, initialize_agent,
    ↳load_tools
from langchain.agents.format_scratchpad import format_log_to_str
from langchain.agents.output_parsers import (
    JSONAgentOutputParser,
    ReActSingleInputOutputParser,
)
from langchain.chains.conversation.memory import ConversationBufferWindowMemory
from langchain.chat_models import ChatOpenAI
from langchain.llms import OpenAI
from langchain.tools import ArxivQueryRun, WikipediaQueryRun, tool
from langchain.tools.render import render_text_description_and_args,
    ↳format_tool_to_openai_function
from langchain.utilities import ArxivAPIWrapper, WikipediaAPIWrapper
from langchain.prompts import MessagesPlaceholder
from langchain.schema import ChatMessage, SystemMessage

from llamp.mp.agents import (
    MPSummaryExpert,
    MPThermoExpert,
    MPElasticityExpert,
    MPDielectricExpert,
    MPPiezoelectricExpert,
    MPMagnetismExpert,
    MPElectronicExpert,
)

load_dotenv()

OPENAI_API_KEY = os.getenv("OPENAI_API_KEY", None)

# OPENAI_GPT_MODEL = "gpt-4-1106-preview"
OPENAI_GPT_MODEL = "gpt-3.5-turbo-1106"
```

```

[ ]: import re

mp_llm = ChatOpenAI(
    temperature=0,
    model="gpt-4-1106-preview",
    openai_api_key=OPENAI_API_KEY,
    openai_organization=None,
    max_retries=5,
    # streaming=True
)

llm = ChatOpenAI(
    temperature=0,
    model="gpt-4",
    openai_api_key=OPENAI_API_KEY,
    openai_organization=None,
    # streaming=True
)

wikipedia = WikipediaQueryRun(api_wrapper=WikipediaAPIWrapper())
arxiv = ArxivQueryRun(api_wrapper=ArxivAPIWrapper())

tools = [
    MPThermoExpert(llm=mp_llm).
    ↪as_tool(agent_kwargs=dict(return_intermediate_steps=False)),
    MPElasticityExpert(llm=mp_llm).
    ↪as_tool(agent_kwargs=dict(return_intermediate_steps=False)),
    MPDielectricExpert(llm=mp_llm).
    ↪as_tool(agent_kwargs=dict(return_intermediate_steps=False)),
    MPMagnetismExpert(llm=mp_llm).
    ↪as_tool(agent_kwargs=dict(return_intermediate_steps=False)),
    MPElectronicExpert(llm=mp_llm).
    ↪as_tool(agent_kwargs=dict(return_intermediate_steps=False)),
    MPSummaryExpert(llm=mp_llm).
    ↪as_tool(agent_kwargs=dict(return_intermediate_steps=False)),
    # arxiv,
    # wikipedia,
]

prompt = hub.pull("hwchase17/react-multi-input-json")
prompt.messages[0].prompt.template = re.sub(
    r"\s+", " ",
    """You are a data-aware agent that can consult materials-related
    data through Materials Project (MP) database, arXiv, and Wikipedia. Ask
    user to clarify their queries if needed. Please note that you don't have
    direct control over MP but through multiple assistant agents to help you.
    You need to provide complete context in the input for them to do their job.
  """
)

```

```

        """).replace("\n", " ") + prompt.messages[0].prompt.template

prompt = prompt.partial(
    tools=render_text_description_and_args(tools),
    tool_names=", ".join([t.name for t in tools]),
)

agent = (
    {
        "input": lambda x: x["input"],
        "agent_scratchpad": lambda x:
↪format_log_to_str(x["intermediate_steps"]),
    }
    | prompt
    | llm.bind(stop=["Observation"])
    # / map_reduce_chain # TODO: Add map-reduce after LLM
    | JSONAgentOutputParser()
)

conversational_memory = ConversationBufferWindowMemory(
    memory_key='chat_history',
    k=5,
    return_messages=True
)

agent_kwargs = {
    "handle_parsing_errors": True,
    "extra_prompt_messages": [
        MessagesPlaceholder(variable_name="chat_history"),
        # SystemMessage(content=re.sub(
        #     r"\s+", " ",
        #     """"You are a helpful data-aware agent that can consult
↪materials-related
        #     data through Materials Project (MP) database, arXiv, and
↪Wikipedia. Ask
        #     user to clarify their queries if needed. Please note that you
↪don't have
        #     direct control to MP but through multiple assistant agents to
↪help you.
        #     You need to provide complete context for them to do their job.
        #     """).replace("\n", " ")
        # )
    ],
    # "early_stopping_method": 'generate',
    # "extra_prompt_messages":
    # )

```

```

}

agent_executor = initialize_agent(
    agent=AgentType.STRUCTURED_CHAT_ZERO_SHOT_REACT_DESCRIPTION,
    tools=tools,
    llm=llm,
    verbose=True,
    max_iterations=10,
    memory=conversational_memory,
    agent_kwargs=agent_kwargs,
    handle_parsing_errors=True,
)

# agent_executor = initialize_agent(
#     tools=tools,
#     llm=llm,
#     agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION,
#     verbose=True,
#     max_iterations=5,
# )

```

```

/home/cyrus/miniconda3/envs/llamp/lib/python3.11/site-
packages/mp_api/client/mprester.py:230: UserWarning: mpcontribs-client not
installed. Install the package to query MPContribs data, or construct pourbaix
diagrams: 'pip install mpcontribs-client'
    warnings.warn(

```

```

[ ]: agent_executor.invoke({
    "input": "What's the bandgap of CsPbI3?"
})

```

> Entering new AgentExecutor chain...

Thought: The bandgap of a material is a property related to its electronic structure. Therefore, I should use the MPElectronicExpert tool to find this information.

Action:

```
...  
  
{  
  "action": "MPElectronicExpert",  
  "action_input": "What's the bandgap of CsPbI3?"  
}  
...
```

> Entering new AgentExecutor chain...

```
/home/cyrus/miniconda3/envs/llamp/lib/python3.11/site-  
packages/mp_api/client/mprester.py:230: UserWarning: mpcontribs-client not  
installed. Install the package to query MPContribs data, or construct pourbaix  
diagrams: 'pip install mpcontribs-client'
```

```
warnings.warn(  
/home/cyrus/miniconda3/envs/llamp/lib/python3.11/site-  
packages/mp_api/client/mprester.py:230: UserWarning: mpcontribs-client not  
installed. Install the package to query MPContribs data, or construct pourbaix  
diagrams: 'pip install mpcontribs-client'  
warnings.warn(  
Action:  
...  
  
{  
  "action": "search_materials_electronic_structure__get",  
  "action_input": {  
    "formula": "CsPbI3"  
  }  
}  
...{"formula": "CsPbI3"}  
Retrieving ElectronicStructureDoc documents: 0%|          | 0/3 [00:00<?, ?it/  
  s]
```

```

[{'material_id': 'mp-1120768', 'last_updated':
'2023-08-12T09:16:31.142000'}, {'material_id': 'mp-1069538', 'last_updated':
'2023-08-13T07:19:55.398000'}, {'material_id': 'mp-540839', 'last_updated':
'2023-08-13T15:45:51.364000'}]]Action:
...

{
  "action": "search_materials_electronic_structure__get",
  "action_input": {
    "formula": "CsPbI3",
    "fields": "material_id,band_gap"
  }
}
```{"formula": "CsPbI3", "fields": "material_id,band_gap"}
Retrieving ElectronicStructureDoc documents: 0%| | 0/3 [00:00<?, ?it/
↵s]
[{'band_gap': 1.6421000000000001, 'material_id': 'mp-1120768'},
{'band_gap': 1.4785000000000001, 'material_id': 'mp-1069538'}, {'band_gap':
2.5181, 'material_id': 'mp-540839'}]]Action:
...

{
 "action": "Final Answer",
 "action_input": "The band gaps for CsPbI3 are as follows: 1.642 eV for
material mp-1120768, 1.479 eV for material mp-1069538, and 2.518 eV for material
mp-540839."
}
...

> Finished chain.

Observation: {'input': "What's the bandgap of CsPbI3?", 'output':
'The band gaps for CsPbI3 are as follows: 1.642 eV for material mp-1120768,
1.479 eV for material mp-1069538, and 2.518 eV for material mp-540839.'}

```

Thought:The MPElectronicExpert tool provided the bandgap values for different materials of CsPbI3. I can now provide this information to the user.

Action:

...

{

  "action": "Final Answer",

  "action\_input": "The band gaps for CsPbI3 are as follows: 1.642 eV for material mp-1120768, 1.479 eV for material mp-1069538, and 2.518 eV for material mp-540839."

}

...

> Finished chain.

```
[]: {'input': "What's the bandgap of CsPbI3?",
 'chat_history': [HumanMessage(content="What's the stiffest materials with the lowest formation energy in Si-O system?"),
 AIMessage(content='The stiffest material in the Si-O system is SiO2, and it has a formation energy per atom that varies between -3.039 eV/atom and -2.951 eV/atom depending on the specific material entry. The material with the lowest formation energy in the Si-O system is Si2O5, but it does not have available stiffness data. Therefore, the stiffest material with the lowest formation energy in the Si-O system is SiO2. '),
 HumanMessage(content='Could you give me any perovskite materials with high dielectric constant?'),
 AIMessage(content='The perovskite material with a high dielectric constant is TaAgO3 (mp-9890) with a dielectric constant of 1501.07. '),
 HumanMessage(content='Could you give me any perovskite materials with bandgap around 2eV?'),
 AIMessage(content="I'm sorry, but there are no perovskite materials with a bandgap around 2eV found in the current database."),
 HumanMessage(content='Could you give me any perovskite materials with bandgap around 1.2 eV?'),
 AIMessage(content='No perovskite materials with a bandgap around 1.2 eV were found in the current database. ')],
 'output': 'The band gaps for CsPbI3 are as follows: 1.642 eV for material mp-1120768, 1.479 eV for material mp-1069538, and 2.518 eV for material mp-540839.'}
```

[ ]: