

Importing necessary libraries:

```
[34]: import json
import re
import os
os.environ['OPENAI_API_KEY']=""
```

OpenAI key has been hidden but could also be imported from a .env file.

Loading the JSON file and doing some preprocessing:

```
[35]: f = open('news.article.json')
data = json.load(f)
docs=""
# Iterating through the json
# List
for i in data:
    s=i['articleBody']
    s=s.replace("\'\'s", "'s")
    s = re.sub(r'\n+', ' ', s)
    #s=s.replace('\s', '')
    s.replace(r'\n', '')
    docs+=s
```

Loading Wikipedia API through Langchain to also get results from Wikipedia:

```
[36]: from langchain_community.tools import WikipediaQueryRun
from langchain_community.utilities import WikipediaAPIWrapper
api_wrapper=WikipediaAPIWrapper(top_k_results=1,doc_content_chars_max=1000)
wiki=WikipediaQueryRun(api_wrapper=api_wrapper)
```

Dividing the preprocessed data into chunks and storing into FAISS database for fast query search through retriever:

```
[39]: from langchain_community.vectorstores import FAISS
from langchain_openai import OpenAIEmbeddings
from langchain_text_splitters import RecursiveCharacterTextSplitter
from langchain_chroma import Chroma

text_splitter = RecursiveCharacterTextSplitter(chunk_size=3500, chunk_overlap=100)
chunks = text_splitter.split_text(docs)
embeddings = OpenAIEmbeddings()
```

Recursivecharactertextspliter from Langchain was used to divide in chunks with chunk_size=3500. Many different chunk sizes were tried, but this performed the best.

```
[40]: vectordb = FAISS.from_texts(chunks[:500], embeddings)
retriever = vectordb.as_retriever()
```

Many databases were tried like chromadb, astradb, pinecone and FAISS. FAISS outperformed and best suited our needs.

Loading retriever tool from Langchain for efficient query search:

```
[41]: from langchain.tools.retriever import create_retriever_tool
retriever_tool=create_retriever_tool(retriever,"vector_search",
    "Search for information about Israel-Hamas War. For any questions about Israel-Hamas War, you must use this tool!")
```

Passing the list of tools that LLM can use:

```
[42]: tools=[wiki,retriever_tool]
```

Loading OpenAI Model:

```
[43]: from langchain_openai import ChatOpenAI

llm = ChatOpenAI(model="gpt-3.5-turbo-0125", temperature=0)
```

Import a generic RAG template from Langchain hub:

```
[44]: from langchain import hub
# Get the prompt to use - you can modify this!
prompt = hub.pull("hwchase17/openai-functions-agent")
prompt.messages

[44]: [SystemMessagePromptTemplate(prompt=PromptTemplate(input_variables=[], template='You are a helpful assistant')),
MessagesPlaceholder(variable_name='chat_history', optional=True),
HumanMessagePromptTemplate(prompt=PromptTemplate(input_variables=['input'], template='{input}')),
MessagesPlaceholder(variable_name='agent_scratchpad')]
```

Creating agents:

```
[45]: ### Agents
from langchain.agents import create_openai_tools_agent
agent=create_openai_tools_agent(llm,tools,prompt)
```

Creating AgentExecutor and providing the set of tools while verbose is true:

```
[46]: ## Agent Executor
from langchain.agents import AgentExecutor
agent_executor=AgentExecutor(agent=agent,tools=tools,verbose=True)
agent_executor

[46]: AgentExecutor(verbose=True, agent=RunnableMultiActionAgent(runnable=RunnableAssign(mapper={
  agent_scratchpad: RunnableLambda(lambda x: format_to_openai_tool_messages(x['intermediate_steps']))
})), ChatPromptTemplate(input_variables=['agent_scratchpad', 'input'], input_types={'chat_history': typing.List[typing.Union[langchain_core.messages.ai.AIMessage, langchain_core.messages.human.HumanMessage, langchain_core.messages.chat.ChatMessage, langchain_core.messages.system.SystemMessage, langchain_core.messages.function.FunctionMessage, langchain_core.messages.tool.ToolMessage]], 'agent_scratchpad': typing.List[typing.Union[langchain_core.messages.ai.AIMessage, langchain_core.messages.human.HumanMessage, langchain_core.messages.chat.ChatMessage, langchain_core.messages.system.SystemMessage, langchain_core.messages.function.FunctionMessage, langchain_core.messages.tool.ToolMessage]]], metadata={'lc_hub_owner': 'hwchase17', 'lc_hub_repo': 'openai-functions-agent', 'lc_hub_commit_hash': 'a1655024b06afbd95d17449f21316291e0726f13dcfaf990cc0d18087ad689a5'}), messages=[SystemMessagePromptTemplate(prompt=PromptTemplate(input_variables=[], template='You are a helpful assistant')), MessagesPlaceholder(variable_name='chat_history', optional=True), HumanMessagePromptTemplate(prompt=PromptTemplate(input_variables=['input'], template='{input}')), MessagesPlaceholder(variable_name='agent_scratchpad'))])
| RunnableBinding(bound=ChatOpenAI(client=<openai.resources.chat.completions.Completions object at 0x7f97bc9e3fa0>, async_client=<openai.resources.chat.completions.AsyncCompletions object at 0x7f97be2b5d80>, model_name='gpt-3.5-turbo-0125', temperature=0.0, openai_api_key=SecretStr('*****'), openai_proxy=''), kwargs={'tools': [{'type': 'function', 'function': {'name': 'wikipedia', 'description': 'A wrapper around Wikipedia. Useful for when you need to answer general questions about people, places, companies, facts, historical events, or other subjects. Input should be a search query.', 'parameters': {'properties': {'__arg1': {'title': '__arg1', 'type': 'string'}}, 'required': ['__arg1'], 'type': 'object'}}}], {'type': 'function', 'function': {'name': 'vector_search', 'description': 'Search for information about Israel-Hamas War. For any questions about Israel-Hamas War, you must use this tool!', 'parameters': {'type': 'object', 'properties': {'query': {'description': 'query to look up in retriever', 'type': 'string'}}, 'required': ['query']}}]}))
```

Sample Output:

```
[47]: agent_executor.invoke({"input": "What happend at the AI-Shifa Hospital"})
```

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

```
Invoking: `vector_search` with `{'query': 'AI-Shifa Hospital'}`
```

that more than one-third of the city's hospitals are no longer operating and Israel's bombing continues, health care professionals fear the worst. "The health system here is in its last throes before it completely boots down. If the electricity goes, that's it. It just becomes a mass grave," says Dr. Ghassan Abu-Sittah, a British-Palestinian plastic and reconstructive surgeon who has been working at Al-Shifa hospital over the last two weeks. "There's no hospital, if there's no electricity." Read More: "Our Death Is Pending." Stories of Loss and Grief From Gaza Right now, his sense is that it will be "days, rather than weeks," until Al-Shifa runs out of fuel needed to keep the hospital running. Palestine's Ministry of Health said Tuesday that hospital generators will stop running in 48 hours, and aid workers tell TIME that the city is expected to run out of fuel on Wednesday evening. More From TIME The situation is particularly dire for neonatal babies. Dr. Hatem Edhair, the head of the neonatal intensive care unit at Nasser Medical Complex in Khan Younis, fears that electricity shutting off will mean five babies dependent on ventilators will die. "If there is no electricity, it means the end of their life... because oxygen will not be available," he says. Dr. Ahmed Mhanna, manager of Al-Awda hospitals in northern Gaza, said Monday that the hospital only had enough fuel to run another three to four days. They have been relying on two generators that have been consuming more than 13 liters per hour, he says. "If there's no fuel, it means the generator will stop. If the generator will stop, the hospital will

```
> Finished chain.
```

```
[47]: {'input': 'What happend at the AI-Shifa Hospital',
'output': "The Al-Shifa Hospital in Gaza has been facing a critical situation due to the ongoing conflict between Israel and Hamas. The hospital is struggling to operate as it is running out of fuel needed to keep essential services running. The electricity supply is at risk, which is crucial for the hospital's operations, especially for neonatal babies dependent on ventilators. The hospital is overcrowded, operating well beyond its capacity, and facing shortages of medical supplies and equipment.\n\nThe conflict has resulted in a high number of casualties, with more than 6,400 people killed and over 17,000 injured in Gaza. The hospitals, including Al-Shifa, are overwhelmed with patients, and the healthcare system is on the brink of collapse. The situation is exacerbated by the lack of clean water, inadequate sanitation, and the risk of infectious diseases spreading in the crowded hospital environment.\n\nDespite the challenges, healthcare professionals at Al-Shifa Hospital are dedicated to continuing their mission of caring for patients, even in the face of constant bombings and threats to their safety. The hospital staff are working tirelessly to provide medical care under extremely difficult circumstances.\n\nThe international community has expressed concerns about the humanitarian crisis in Gaza and the urgent need for humanitarian aid to support the healthcare system and provide relief to the affected population. Efforts are being made to provide medical supplies and support to hospitals in Gaza, but more assistance is required to address the growing needs of the healthcare system in the region."}
```

As you can see, since the model was able to find relevant data from the database, it outputs that information. If unable to find related data in the database, it will output results from Wikipedia as shown in the next example.

Example2:

```
[48]: agent_executor.invoke({"input":"What is FIFA"})
```

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

```
Invoking: `wikipedia` with `FIFA`
```

```
Page: FIFA
```

```
Summary: The Fédération Internationale de Football Association (French for 'International Federation of Association Football'; abbreviated as FIFA and pronounced in English as FEE-fa) is an international self-regulatory governing body of association football, beach soccer, and futsal. It was founded in 1904 to oversee international competition among the national associations of Belgium, Denmark, France, Germany, the Netherlands, Spain (represented by the Madrid Football Club), Sweden, and Switzerland. Headquartered in Zürich, Switzerland, its membership now comprises 211 national associations. These national associations must also be members of one of the six regional confederations into which the world is divided: CAF (Africa), AFC (Asia and Australasia), UEFA (Europe), CONCACAF (North & Central America and the Caribbean), OFC (Oceania), and CONMEBOL (South America). FIFA outlines several objectives in its organizational statutes, including growing association football internationally. FIFA stands for Fédération Internationale de Football Association, which is the international governing body of association football, beach soccer, and futsal. It was founded in 1904 and is headquartered in Zurich, Switzerland. FIFA oversees international competitions among national football associations and has 211 member associations. The organization is divided into six regional confederations: CAF (Africa), AFC (Asia and Australia), UEFA (Europe), CONCACAF (North & Central America and the Caribbean), OFC (Oceania), and CONMEBOL (South America). FIFA has various objectives outlined in its organizational statutes, including growing association football internationally.
```

```
> Finished chain.
```

```
[48]: {'input': 'What is FIFA',  
      'output': 'FIFA stands for Fédération Internationale de Football Association, which is the international governing body of association football, beach soccer, and futsal. It was founded in 1904 and is headquartered in Zurich, Switzerland. FIFA oversees international competitions among national football associations and has 211 member associations. The organization is divided into six regional confederations: CAF (Africa), AFC (Asia and Australia), UEFA (Europe), CONCACAF (North & Central America and the Caribbean), OFC (Oceania), and CONMEBOL (South America). FIFA has various objectives outlined in its organizational statutes, including growing association football internationally.'}
```