



# Combine agents and vector stores

This notebook covers how to combine agents and vectorstores. The use case for this is that you've ingested your data into a vectorstore and want to interact with it in an agentic manner.

The recommended method for doing so is to create a RetrievalQA and then use that as a tool in the overall agent. Let's take a look at doing this below. You can do this with multiple different vectordbs, and use the agent as a way to route between them. There are two different ways of doing this - you can either let the agent use the vectorstores as normal tools, or you can set `return_direct=True` to really just use the agent as a router.

## Create the Vectorstore

```
from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.vectorstores import Chroma
from langchain.text_splitter import CharacterTextSplitter
from langchain.llms import OpenAI
from langchain.chains import RetrievalQA
```

```
llm = OpenAI(temperature=0)
```

### API Reference:

- `OpenAIEmbeddings` from `langchain.embeddings.openai`
- `Chroma` from `langchain.vectorstores`
- `CharacterTextSplitter` from `langchain.text_splitter`
- `OpenAI` from `langchain.llms`
- `RetrievalQA` from `langchain.chains`

```
from pathlib import Path

relevant_parts = []
for p in Path(".").absolute().parts:
    relevant_parts.append(p)
```



```

    if relevant_parts[-3:] == ["langchain", "docs", "modules"]:
        break
doc_path = str(Path(*relevant_parts) / "state_of_the_union.txt")

```

```

from langchain.document_loaders import TextLoader

loader = TextLoader(doc_path)
documents = loader.load()
text_splitter = CharacterTextSplitter(chunk_size=1000, chunk_overlap=0)
texts = text_splitter.split_documents(documents)

embeddings = OpenAIEmbeddings()
docsearch = Chroma.from_documents(texts, embeddings,
collection_name="state-of-union")

```

#### API Reference:

- `TextLoader` from `langchain.document_loaders`

Running Chroma using direct local API.  
Using DuckDB in-memory for database. Data will be transient.

```

state_of_union = RetrievalQA.from_chain_type(
    llm=llm, chain_type="stuff", retriever=docsearch.as_retriever()
)

```

```

from langchain.document_loaders import WebBaseLoader

```

#### API Reference:

- `WebBaseLoader` from `langchain.document_loaders`

```

loader = WebBaseLoader("https://beta.ruff.rs/docs/faq/")

```

```

docs = loader.load()
ruff_texts = text_splitter.split_documents(docs)

```

```
ruff_db = Chroma.from_documents(ruff_texts, embeddings,
collection_name="ruff")
ruff = RetrievalQA.from_chain_type(
    llm=llm, chain_type="stuff", retriever=ruff_db.as_retriever()
)
```

Running Chroma using direct local API.  
Using DuckDB in-memory for database. Data will be transient.

## Create the Agent

```
# Import things that are needed generically
from langchain.agents import initialize_agent, Tool
from langchain.agents import AgentType
from langchain.tools import BaseTool
from langchain.llms import OpenAI
from langchain import LLMMathChain, SerpAPIWrapper
```

### API Reference:

- `initialize_agent` from `langchain.agents`
- `Tool` from `langchain.agents`
- `AgentType` from `langchain.agents`
- `BaseTool` from `langchain.tools`
- `OpenAI` from `langchain.llms`

```
tools = [
    Tool(
        name="State of Union QA System",
        func=state_of_union.run,
        description="useful for when you need to answer questions about  
the most recent state of the union address. Input should be a fully formed  
question.",
    ),
    Tool(
        name="Ruff QA System",
        func=ruff.run,
```

```

        description="useful for when you need to answer questions about
ruff (a python linter). Input should be a fully formed question.",
    ),
]

```

```

# Construct the agent. We will use the default agent type here.
# See documentation for a full list of options.
agent = initialize_agent(
    tools, llm, agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION, verbose=True
)

```

```

agent.run(
    "What did biden say about ketanji brown jackson in the state of the
union address?"
)

```

```

> Entering new AgentExecutor chain...
I need to find out what Biden said about Ketanji Brown Jackson in the
State of the Union address.
Action: State of Union QA System
Action Input: What did Biden say about Ketanji Brown Jackson in the
State of the Union address?
Observation: Biden said that Jackson is one of the nation's top legal
minds and that she will continue Justice Breyer's legacy of excellence.
Thought: I now know the final answer
Final Answer: Biden said that Jackson is one of the nation's top legal
minds and that she will continue Justice Breyer's legacy of excellence.

> Finished chain.

```

"Biden said that Jackson is one of the nation's top legal minds and that she will continue Justice Breyer's legacy of excellence."

```
agent.run("Why use ruff over flake8?")
```

> Entering new AgentExecutor chain...

I need to find out the advantages of using ruff over flake8

Action: Ruff QA System

Action Input: What are the advantages of using ruff over flake8?

Observation: Ruff can be used as a drop-in replacement for Flake8 when used (1) without or with a small number of plugins, (2) alongside Black, and (3) on Python 3 code. It also re-implements some of the most popular Flake8 plugins and related code quality tools natively, including isort, yesqa, eradicate, and most of the rules implemented in pyupgrade. Ruff also supports automatically fixing its own lint violations, which Flake8 does not.

Thought: I now know the final answer

Final Answer: Ruff can be used as a drop-in replacement for Flake8 when used (1) without or with a small number of plugins, (2) alongside Black, and (3) on Python 3 code. It also re-implements some of the most popular Flake8 plugins and related code quality tools natively, including isort, yesqa, eradicate, and most of the rules implemented in pyupgrade. Ruff also supports automatically fixing its own lint violations, which Flake8 does not.

> Finished chain.

'Ruff can be used as a drop-in replacement for Flake8 when used (1) without or with a small number of plugins, (2) alongside Black, and (3) on Python 3 code. It also re-implements some of the most popular Flake8 plugins and related code quality tools natively, including isort, yesqa, eradicate, and most of the rules implemented in pyupgrade. Ruff also supports automatically fixing its own lint violations, which Flake8 does not.'

## Use the Agent solely as a router

You can also set `return_direct=True` if you intend to use the agent as a router and just want to directly return the result of the RetrievalQAChain.

Notice that in the above examples the agent did some extra work after querying the RetrievalQAChain. You can avoid that and just return the result directly.

```
tools = [
    Tool(
        name="State of Union QA System",
        func=state_of_union.run,
        description="useful for when you need to answer questions about the most recent state of the union address. Input should be a fully formed question.",
        return_direct=True,
    ),
    Tool(
        name="Ruff QA System",
        func=ruff.run,
        description="useful for when you need to answer questions about ruff (a python linter). Input should be a fully formed question.",
        return_direct=True,
    ),
]
```

```
agent = initialize_agent(
    tools, llm, agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION, verbose=True
)
```

```
agent.run(
    "What did Biden say about Ketanji Brown Jackson in the state of the union address?"
)
```

> Entering new AgentExecutor chain...

I need to find out what Biden said about Ketanji Brown Jackson in the State of the Union address.

Action: State of Union QA System

Action Input: What did Biden say about Ketanji Brown Jackson in the State of the Union address?

Observation: Biden said that Jackson is one of the nation's top legal minds and that she will continue Justice Breyer's legacy of excellence.

> Finished chain.

" Biden said that Jackson is one of the nation's top legal minds and that she will continue Justice Breyer's legacy of excellence."

`agent.run("Why use ruff over flake8?")`

> Entering new AgentExecutor chain...

I need to find out the advantages of using ruff over flake8

Action: Ruff QA System

Action Input: What are the advantages of using ruff over flake8?

Observation: Ruff can be used as a drop-in replacement for Flake8 when used (1) without or with a small number of plugins, (2) alongside Black, and (3) on Python 3 code. It also re-implements some of the most popular Flake8 plugins and related code quality tools natively, including isort, yesqa, eradicate, and most of the rules implemented in pyupgrade. Ruff also supports automatically fixing its own lint violations, which Flake8 does not.

> Finished chain.

' Ruff can be used as a drop-in replacement for Flake8 when used (1) without or with a small number of plugins, (2) alongside Black, and (3) on Python 3 code. It also re-implements some of the most popular Flake8 plugins and related code quality tools natively, including isort, yesqa,

eradicate, and most of the rules implemented in pyupgrade. Ruff also supports automatically fixing its own lint violations, which Flake8 does not.'

## Multi-Hop vectorstore reasoning

Because vectorstores are easily usable as tools in agents, it is easy to use answer multi-hop questions that depend on vectorstores using the existing agent framework

```
tools = [
    Tool(
        name="State of Union QA System",
        func=state_of_union.run,
        description="useful for when you need to answer questions about the most recent state of the union address. Input should be a fully formed question, not referencing any obscure pronouns from the conversation before.",
    ),
    Tool(
        name="Ruff QA System",
        func=ruff.run,
        description="useful for when you need to answer questions about ruff (a python linter). Input should be a fully formed question, not referencing any obscure pronouns from the conversation before.",
    ),
]
```

```
# Construct the agent. We will use the default agent type here.
# See documentation for a full list of options.
agent = initialize_agent(
    tools, llm, agent=AgentType.ZERO_SHOT_REACT_DESCRIPTION, verbose=True
)
```

```
agent.run(
    "What tool does ruff use to run over Jupyter Notebooks? Did the president mention that tool in the state of the union?"
)
```



> Entering new AgentExecutor chain...

I need to find out what tool ruff uses to run over Jupyter Notebooks, and if the president mentioned it in the state of the union.

Action: Ruff QA System

Action Input: What tool does ruff use to run over Jupyter Notebooks?

Observation: Ruff is integrated into nbQA, a tool for running linters and code formatters over Jupyter Notebooks. After installing ruff and nbqa, you can run Ruff over a notebook like so: > nbqa ruff Untitled.html

Thought: I now need to find out if the president mentioned this tool in the state of the union.

Action: State of Union QA System

Action Input: Did the president mention nbQA in the state of the union?

Observation: No, the president did not mention nbQA in the state of the union.

Thought: I now know the final answer.

Final Answer: No, the president did not mention nbQA in the state of the union.

> Finished chain.

'No, the president did not mention nbQA in the state of the union.'