Caching

LangChain provides an optional caching layer for LLMs. This is useful for two reasons:

It can save you money by reducing the number of API calls you make to the LLM provider, if you're often requesting the same completion multiple times. It can speed up your application by reducing the number of API calls you make to the LLM provider.

```
import langchain
from langchain.llms import OpenAI

# To make the caching really obvious, lets use a slower model.
llm = OpenAI(model_name="text-davinci-002", n=2, best_of=2)
```

In Memory Cache

```
from langchain.cache import InMemoryCache
langchain.llm_cache = InMemoryCache()

# The first time, it is not yet in cache, so it should take longer
llm.predict("Tell me a joke")
```

```
    CPU times: user 35.9 ms, sys: 28.6 ms, total: 64.6 ms
    Wall time: 4.83 s
```

```
    "\n\nWhy couldn't the bicycle stand up by itself? It was...two tired!"
```

```
# The second time it is, so it goes faster
llm.predict("Tell me a joke")
```

```
    CPU times: user 238 µs, sys: 143 µs, total: 381 µs
    Wall time: 1.76 ms


    '\n\nWhy did the chicken cross the road?\n\nTo get to the other side.'
```

SQLite Cache

```
rm .langchain.db
```

```
# We can do the same thing with a SQLite cache
from langchain.cache import SQLiteCache
langchain.llm_cache = SQLiteCache(database_path=".langchain.db")
```

```
# The first time, it is not yet in cache, so it should take longer
llm.predict("Tell me a joke")
```

```
    CPU times: user 17 ms, sys: 9.76 ms, total: 26.7 ms
    Wall time: 825 ms


    '\n\nWhy did the chicken cross the road?\n\nTo get to the other side.'
```

```
# The second time it is, so it goes faster
llm.predict("Tell me a joke")
```

```
    CPU times: user 2.46 ms, sys: 1.23 ms, total: 3.7 ms
    Wall time: 2.67 ms


    '\n\nWhy did the chicken cross the road?\n\nTo get to the other side.'
```

Optional Caching in Chains

You can also turn off caching for particular nodes in chains. Note that because of certain interfaces, its often easier to construct the chain first, and then edit the LLM afterwards.

As an example, we will load a summarizer map-reduce chain. We will cache results for the map-step, but then not freeze it for the combine step.

```
llm = OpenAI(model_name="text-davinci-002")
no_cache_llm = OpenAI(model_name="text-davinci-002", cache=False)
```

```
from langchain.text_splitter import CharacterTextSplitter
from langchain.chains.mapreduce import MapReduceChain

text_splitter = CharacterTextSplitter()
```

```
with open('../../../state_of_the_union.txt') as f:
    state_of_the_union = f.read()
texts = text_splitter.split_text(state_of_the_union)
```

```
from langchain.docstore.document import Document
docs = [Document(page_content=t) for t in texts[:3]]
from langchain.chains.summarize import load_summarize_chain
```

```
chain = load_summarize_chain(llm, chain_type="map_reduce", reduce_llm=no_cache_llm)
```

```
chain.run(docs)
```

```
    CPU times: user 452 ms, sys: 60.3 ms, total: 512 ms
    Wall time: 5.09 s


    '\n\nPresident Biden is discussing the American Rescue Plan and the Bipartisan Infrastructure Law, which will create jobs and help Americans. He also talks about his vision for America, which includes investing in education and infrastructure. In response to Russian aggression in Ukraine, the United States is joining with European allies to impose sanctions and isolate Russia. American forces are being mobilized to protect NATO countries in the event that Putin decides to keep moving west. The Ukrainians are bravely fighting back, but
```

```
the next few weeks will be hard for them. Putin will pay a high price for his actions in the long run.
Americans should not be alarmed, as the United States is taking action to protect its interests and allies.'
```

When we run it again, we see that it runs substantially faster but the final answer is different. This is due to caching at the map steps, but not at the reduce step.

```
chain.run(docs)
```

```
    CPU times: user 11.5 ms, sys: 4.33 ms, total: 15.8 ms
    Wall time: 1.04 s


    '\n\nPresident Biden is discussing the American Rescue Plan and the Bipartisan Infrastructure Law, which
will create jobs and help Americans. He also talks about his vision for America, which includes investing in
education and infrastructure.'
```

```
rm .langchain.db sqlite.db
```