Entity Memory with SQLite storage

In this walkthrough we'll create a simple conversation chain which uses ConversationEntityMemory backed by a SqliteEntityStore.

```python
from langchain.chains import ConversationChain
from langchain.llms import OpenAI
from langchain.memory import ConversationEntityMemory
from langchain.memory.entity import SQLiteEntityStore
from langchain.memory.prompt import ENTITY_MEMORY_CONVERSATION_TEMPLATE
```

```python
entity_store = SQLiteEntityStore()
llm = OpenAI(temperature=0)
memory = ConversationEntityMemory(llm=llm, entity_store=entity_store)
conversation = ConversationChain(
    llm=llm,
    prompt=ENTITY_MEMORY_CONVERSATION_TEMPLATE,
    memory=memory,
    verbose=True,
)
```

Notice the usage of `EntitySqliteStore` as parameter to `entity_store` on the `memory` property.

```python
conversation.run("Deven & Sam are working on a hackathon project")
```

```
> Entering new ConversationChain chain...
Prompt after formatting:
You are an assistant to a human, powered by a large language model trained by OpenAI.

You are designed to be able to assist with a wide range of tasks, from answering simple questions to
providing in-depth explanations and discussions on a wide range of topics. As a language model, you are able
to generate human-like text based on the input you receive, allowing you to engage in natural-sounding
conversations and provide responses that are coherent and relevant to the topic at hand.

You are constantly learning and improving, and your capabilities are constantly evolving. You are able to
process and understand large amounts of text, and can use this knowledge to provide accurate and informative
responses to a wide range of questions. You have access to some personalized information provided by the
human in the Context section below. Additionally, you are able to generate your own text based on the input
you receive, allowing you to engage in discussions and provide explanations and descriptions on a wide range
of topics.

Overall, you are a powerful tool that can help with a wide range of tasks and provide valuable insights
and information on a wide range of topics. Whether the human needs help with a specific question or just
wants to have a conversation about a particular topic, you are here to assist.

Context:
{'Deven': 'Deven is working on a hackathon project with Sam.', 'Sam': 'Sam is working on a hackathon
project with Deven.'}

Current conversation:

Last line:
Human: Deven & Sam are working on a hackathon project
You:
```

```
    > Finished chain.




    ' That sounds like a great project! What kind of project are they working on?'
```

```
conversation.memory.entity_store.get("Deven")
```

```
    'Deven is working on a hackathon project with Sam.'
```

```
conversation.memory.entity_store.get("Sam")
```

```
    'Sam is working on a hackathon project with Deven.'
```