🏠 Modules More Memory

Memory types Conversation Knowledge Graph

Conversation Knowledge Graph

This type of memory uses a knowledge graph to recreate memory.

Using memory with LLM

```
from langchain.memory import ConversationKGMemory
from langchain_openai import OpenAI
```

```
llm = OpenAI(temperature=0)
memory = ConversationKGMemory(llm=llm)
memory.save_context({"input": "say hi to sam"}, {"output": "who is sam"})
memory.save_context({"input": "sam is a friend"}, {"output": "okay"})
memory.load_memory_variables({"input": "who
is sam"})
```

```
{'history': 'On Sam: Sam is friend.'}
```

We can also get the history as a list of messages (this is useful if you are using this with a chat model).

```
memory = ConversationKGMemory(llm=llm,
return_messages=True)
memory.save_context({"input": "say hi to
sam"}, {"output": "who is sam"})
memory.save_context({"input": "sam is a
friend"}, {"output": "okay"})
```

```
memory.load_memory_variables({"input": "who
is sam"})
```

```
{'history': [SystemMessage(content='On Sam:
Sam is friend.', additional_kwargs={})]}
```

We can also more modularly get current entities from a new message (will use previous messages as context).

```
memory.get_current_entities("what's Sams favorite color?")
```

```
['Sam']
```

We can also more modularly get knowledge triplets from a new message (will use previous messages as context).

```
memory.get_knowledge_triplets("her favorite color is red")
```

```
[KnowledgeTriple(subject='Sam', predicate='favorite color', object_='red')]
```

Using in a chain

Let’s now use this in a chain!

```
llm = OpenAI(temperature=0)
from langchain.chains import ConversationChain
from langchain.prompts.prompt import PromptTemplate
```

```
template = """The following is a friendly
conversation between a human and an AI. The
AI is talkative and provides lots of specific
details from its context.
If the AI does not know the answer to a
question, it truthfully says it does not
know. The AI ONLY uses information contained
in the "Relevant Information" section and
does not hallucinate.

Relevant Information:

{history}

Conversation:
Human: {input}
AI:"""
prompt = PromptTemplate(input_variables=
["history", "input"], template=template)
conversation_with_kg = ConversationChain(
    llm=llm, verbose=True, prompt=prompt,
memory=ConversationKGMemory(llm=llm)
)
```

```
conversation_with_kg.predict(input="Hi,
what's up?")
```

```


> Entering new ConversationChain chain...
Prompt after formatting:
The following is a friendly conversation
between a human and an AI. The AI is
talkative and provides lots of specific
details from its context.
If the AI does not know the answer to a
question, it truthfully says it does not
know. The AI ONLY uses information contained
in the "Relevant Information" section and
does not hallucinate.

Relevant Information:




Conversation:
Human: Hi, what's up?
AI:

> Finished chain.
```

```
" Hi there! I'm doing great. I'm currently in
the process of learning about the world
around me. I'm learning about different
```

```
cultures, languages, and customs. It's really
fascinating! How about you?"
```

```
conversation_with_kg.predict(
    input="My name is James and I'm helping
Will. He's an engineer."
)
```

```



> Entering new ConversationChain chain...
Prompt after formatting:
The following is a friendly conversation
between a human and an AI. The AI is
talkative and provides lots of specific
details from its context.
If the AI does not know the answer to a
question, it truthfully says it does not
know. The AI ONLY uses information contained
in the "Relevant Information" section and
does not hallucinate.

Relevant Information:



Conversation:
Human: My name is James and I'm helping Will.
```

```
He's an engineer.
AI:

> Finished chain.
```

```
" Hi James, it's nice to meet you. I'm an AI
and I understand you're helping Will, the
engineer. What kind of engineering does he
do?"
```

```
conversation_with_kg.predict(input="What do
you know about Will?")
```

```



> Entering new ConversationChain chain...
Prompt after formatting:
The following is a friendly conversation
between a human and an AI. The AI is
talkative and provides lots of specific
details from its context.
If the AI does not know the answer to a
question, it truthfully says it does not
know. The AI ONLY uses information contained
in the "Relevant Information" section and
does not hallucinate.
```

```
Relevant Information:

On Will: Will is an engineer.

Conversation:
Human: What do you know about Will?
AI:

> Finished chain.
```

```
' Will is an engineer.'
```