Hi, how can I help you?

Can you show me a examples of passing the chat\_history with ConversationalRetrievalChain?

👨‍💻

🤖

Looks like what you are looking for can be found in full [here](https://python.langchain.com/en/latest/use_cases/evaluation/huggingface_datasets.html)

Sure! Here's an example of how to pass the chat\_history parameter with ConversationalRetrievalChain:

from langchain.chains import ConversationalRetrievalChain

from langchain.retrievers import BM25Retriever

retriever = BM25Retriever()

chain = ConversationalRetrievalChain(retriever=retriever, chat\_history=["Hello, how are you?", "I'm doing well, thanks for asking!"])

response = chain.apply("What's your favorite color?")

print(response)

In this example, we're creating a ConversationalRetrievalChain with a BM25Retriever and passing in a chat\_history parameter with two previous messages. Then, we're using the chain to respond to a new message ("What's your favorite color?") and printing the response.

Let me know if you have any other questions!

Verified Sources:

[1. python.langchain.com/en/latest/use\_cases/evaluation/huggingface\_datasets.html](https://python.langchain.com/en/latest/use_cases/evaluation/huggingface_datasets.html" \t "_self)

👍👎

When I have 2nd chat\_history, how to continue with 3rd conversation with both first and second chat history?

👨‍💻

🤖

Sorry, I couldn't find any information in the documentation about that. Expect answer to be less accurate

To continue a conversation with both the first and second chat history, you can simply pass in a list that contains both chat histories as the **chat\_history** parameter when creating the ConversationalRetrievalChain. Here's an example:

from langchain.chains import ConversationalRetrievalChain

from langchain.retrievers import BM25Retriever

retriever = BM25Retriever()

chat\_history\_1 = ["Hello, how are you?", "I'm doing well, thanks for asking!"]

chat\_history\_2 = ["What's your favorite color?", "My favorite color is blue."]

chain = ConversationalRetrievalChain(retriever=retriever, chat\_history=[chat\_history\_1, chat\_history\_2])

response = chain.apply("What do you like to do for fun?")

print(response)

In this example, we're creating a ConversationalRetrievalChain with a BM25Retriever and passing in a **chat\_history** parameter that contains two chat histories. Then, we're using the chain to respond to a new message ("What do you like to do for fun?") and printing the response.

Let me know if you have any other questions!

👍👎

Will chart history append to the list of chat history when I get new result?

👨‍💻

🤖

Looks like what you are looking for can be found in full [here](https://python.langchain.com/en/latest/reference/modules/agents.html)

Yes, when you pass in a **chat\_history** parameter to a ConversationalRetrievalChain, any new messages that you apply to the chain will be appended to the existing chat history list. So if you create a chain with an initial chat history of **["Hello, how are you?"]**, and then apply a new message of **"I'm doing well, thanks for asking!"**, the resulting chat history will be **["Hello, how are you?", "I'm doing well, thanks for asking!"]**.

Let me know if you have any other questions!

Verified Sources:

[1. python.langchain.com/en/latest/modules/chains/index\_examples/chat\_vector\_db.html](https://python.langchain.com/en/latest/modules/chains/index_examples/chat_vector_db.html" \t "_self)

[2. python.langchain.com/en/latest/reference/modules/agents.html](https://python.langchain.com/en/latest/reference/modules/agents.html" \t "_self)

👍👎

I did this to make llm chain; chain = ConversationalRetrievalChain.from\_llm(llm=chat, retriever=vectorstore.as\_retriever(), condense\_question\_prompt=CONDENSE\_PROMPT, qa\_prompt=QA\_PROMPT, ) then; chat\_history = [] question = "Please provide hedlines related to NVDA." result = chain({"question": question, "chat\_history": chat\_history}) I don't get chat history. What is the problem?

👨‍💻

🤖

Looks like what you are looking for can be found in full [here](https://python.langchain.com/en/latest/gallery.html)

Based on the code you provided, it looks like you're initializing the **chat\_history** variable as an empty list (**chat\_history = []**). This means that there won't be any chat history to display when you call **chain({"question": question, "chat\_history": chat\_history})**.

To see the chat history, you'll need to pass in a list of previous messages as the **chat\_history** parameter. For example, you could initialize **chat\_history** with a list of previous messages like this:

chat\_history = ["Hello, how can I help you today?", "I'm looking for news about NVDA."]

Then, when you call **chain({"question": question, "chat\_history": chat\_history})**, the resulting chat history will include these previous messages along with any new messages generated by the chain.

Let me know if that helps or if you have any other questions!

Verified Sources:

[1. python.langchain.com/en/latest/modules/chains/index\_examples/chat\_vector\_db.html](https://python.langchain.com/en/latest/modules/chains/index_examples/chat_vector_db.html" \t "_self)

[2. python.langchain.com/en/latest/modules/memory/getting\_started.html](https://python.langchain.com/en/latest/modules/memory/getting_started.html" \t "_self)

[3. python.langchain.com/en/latest/gallery.html](https://python.langchain.com/en/latest/gallery.html" \t "_self)

👍👎

What about the answer I get from this result? Do they store in the chat\_history list afte run this code?

👨‍💻

🤖

Looks like what you are looking for can be found in full [here](https://python.langchain.com/en/latest/modules/memory/getting_started.html)

No, the answer generated by the chain will not automatically be added to the **chat\_history** list. If you want to include the answer in the chat history, you'll need to manually add it to the list after you receive the response from the chain. For example:

chat\_history = ["Hello, how can I help you today?", "I'm looking for news about NVDA."]

question = "Please provide headlines related to NVDA."

result = chain({"question": question, "chat\_history": chat\_history})

chat\_history.append(result["answer"])

In this example, we're initializing **chat\_history** with two previous messages, then using the chain to respond to a new message ("Please provide headlines related to NVDA."). After we receive the response from the chain (**result**), we're manually adding the answer to the **chat\_history** list using **chat\_history.append(result["answer"])**.

Let me know if that helps or if you have any other questions!

Verified Sources:

[1. python.langchain.com/en/latest/modules/memory/getting\_started.html](https://python.langchain.com/en/latest/modules/memory/getting_started.html" \t "_self)

[2. python.langchain.com/en/latest/use\_cases/code.html](https://python.langchain.com/en/latest/use_cases/code.html" \t "_self)

👍👎

In real case use, how to pile up the chat history both questions and answers to keep having the conversation?

👨‍💻

🤖

Looks like what you are looking for can be found in full [here](https://python.langchain.com/en/latest/modules/agents/agents/examples/chat_conversation_agent.html)

To pile up the chat history in a real case use, you would need to store the chat history somewhere, such as in a database or a file. Then, when a new message comes in, you would retrieve the existing chat history, add the new message to the list, and pass the updated chat history to the ConversationalRetrievalChain. Here's an example of how you might do this with a file:

import json

# Load existing chat history from file

with open("chat\_history.json", "r") as f:

chat\_history = json.load(f)

# Add new message to chat history

new\_message = "What's the weather like today?"

chat\_history.append(new\_message)

# Pass updated chat history to ConversationalRetrievalChain

response = chain({"question": new\_message, "chat\_history": chat\_history})

# Save updated chat history to file

with open("chat\_history.json", "w") as f:

json.dump(chat\_history, f)

In this example, we're loading the existing chat history from a JSON file, adding a new message to the list, passing the updated chat history to the ConversationalRetrievalChain, and then saving the updated chat history back to the file. You could also use a database to store the chat history, and retrieve and update the chat history using SQL queries.

Let me know if you have any other questions!

Verified Sources:

[1. python.langchain.com/en/latest/modules/memory/examples/postgres\_chat\_message\_history.html](https://python.langchain.com/en/latest/modules/memory/examples/postgres_chat_message_history.html" \t "_self)

[2. python.langchain.com/en/latest/modules/agents/agents/examples/chat\_conversation\_agent.html](https://python.langchain.com/en/latest/modules/agents/agents/examples/chat_conversation_agent.html" \t "_self)

👍👎

from langchain.chains import ConversationalRetrievalChain from langchain.retrievers import BM25Retriever from langchain.memories import ConversationBufferMemory from langchain.prompts import PromptTemplate retriever = BM25Retriever() memory = ConversationBufferMemory(memory\_key="chat\_history") question\_prompt = PromptTemplate( input\_variables=["context\_str", "question"], template="Given the context information and not prior knowledge, answer the question in Italian: {question}" ) chain = ConversationalRetrievalChain(retriever=retriever, memory=memory, question\_prompt=question\_prompt) response = chain.apply("What do you like to do for fun?") print(response)

When added the memory, it will remember the conversation right?

👨‍💻

🤖

Looks like what you are looking for can be found in full [here](https://python.langchain.com/en/latest/modules/memory/types/buffer.html)

Yes, when memory is added to a conversation chain or LLMChain, it can remember the conversation history. The ConversationBufferMemory class is an example of a memory class that allows for storing messages and extracting them later as a string or list of messages. The keys in the PromptTemplate and the ConversationBufferMemory must match up for the memory to work correctly.

Verified Sources:

[1. python.langchain.com/en/latest/modules/memory/types/buffer.html](https://python.langchain.com/en/latest/modules/memory/types/buffer.html" \t "_self)

[2. python.langchain.com/en/latest/modules/memory/examples/adding\_memory.html](https://python.langchain.com/en/latest/modules/memory/examples/adding_memory.html" \t "_self)

👍👎

Do I still need to add chat\_history list I asked you before? will memory takes care of piling the chat history?

👨‍💻

🤖

Looks like what you are looking for can be found in full [here](https://python.langchain.com/en/latest/modules/memory/examples/redis_chat_message_history.html)

Yes, when using the ConversationBufferMemory class, it will automatically store the chat history for you. You don't need to manually create a chat\_history list. The chat\_history variable in the PromptTemplate matches up with the dynamic key name in the ConversationBufferMemory, so as long as they match, the memory will work correctly.

Verified Sources:

[1. python.langchain.com/en/latest/modules/memory/examples/redis\_chat\_message\_history.html](https://python.langchain.com/en/latest/modules/memory/examples/redis_chat_message_history.html" \t "_self)

[2. python.langchain.com/en/latest/modules/memory/examples/agent\_with\_](https://python.langchain.com/en/latest/modules/memory/examples/agent_with_memory.html" \t "_self)