# 工具链的理解与应用

在Langchain 中，“链”的概念是最经常使用的。这些”链”其实就是由一系列工具链构成的，每一个工具都可以视为整个链中的一个环节。这些环节可能非常简单，例如将一个提示模板和一个大型语言模型链接起来，形成一个大型语言模型链（LLMChains）。然而，也可能更加复杂，例如在整个流程中，通过多个环节进行多个步骤的链接。这可能还包括多个大型语言模型以及各种不同的实用工具等。在工具链中，一个链的输出将成为下一个链的输入，这就形成了一个输入输出的链式流程。例如，你可能会从大型语言模型的输出中提取某些内容，将其作为Wolfram Alpha查询的输入，然后将查询结果带回，并再次通过大型模型生成将返回给用户的响应。这就是一个典型的工具链的示例。

#### 常见工具链的功能与应用

在实际的应用中，一些常见的工具链如APIChain、ConversationalRetrievalQA等已经被封装好了。

APIChain使得大型语言模型可以与API进行交互，以获取相关的信息。构建该链时，需要提供一个与所提供的API文档相关的问题。

ConversationalRetrievalQA链在检索问答链的基础上提供了一个聊天历史组件。它首先将聊天历史（要么明确传入，要么从提供的内存中检索）和问题合并成一个独立的问题，然后从检索器中查找相关的文档，最后将这些文档和问题传递给一个问答链，以返回响应。

对于需要对多个文档进行文档合并的任务，我们可以使用文档合并链，如MapReduceDocumentsChain或StuffDocumentsChain等。

对于需要从同一段落中提取多个实体及其属性的任务，我们可以使用提取链。

还有一些专门设计用来满足特定需求的链，如ConstitutionalChain，这是一个保证大型语言模型输出遵循一定宪法原则的链，通过设定特定的规则和指导方针，使得生成的内容符合这些原则，从而提供更受控、符合伦理和上下文适当的回应。

#### 工具链的使用方法

这些工具链的使用方法通常是先使用类方法实例化，然后通过run方法调用，输出结果是一个字符串，然后将这个字符串传递给下一个链。类方法通常以”from”和下划线开始，比较常见的有from\_llm()和from\_chain\_type()，他们都接受外部的数据来源作为参数。

下面以SQLDatabaseChain为例子，看看如何使用它。SQLDatabaseChain就是一个通过from\_llm()方法实例化的链，它用于回答SQL数据库上的问题。

from langchain import OpenAI, SQLDatabase, SQLDatabaseChain  
  
db = SQLDatabase.from\_uri("sqlite:///../../../../notebooks/Chinook.db")  
llm = OpenAI(temperature=0, verbose=True)  
  
db\_chain = SQLDatabaseChain.from\_llm(llm, db, verbose=True)  
  
db\_chain.run("How many employees are there?")

运行的结果是：

> Entering new SQLDatabaseChain chain...  
 How many employees are there?  
 SQLQuery:  
  
 /workspace/langchain/langchain/sql\_database.py:191: SAWarning: Dialect sqlite+pysqlite does \*not\* support Decimal objects natively, and SQLAlchemy must convert from floating point - rounding errors and other issues may occur. Please consider storing Decimal numbers as strings or integers on this platform for lossless storage.  
 sample\_rows = connection.execute(command)  
  
  
 SELECT COUNT(\*) FROM "Employee";  
 SQLResult: [(8,)]  
 Answer:There are 8 employees.  
 > Finished chain.  
  
  
  
  
  
 'There are 8 employees.'