# 2.2.1 提示模板

实际上, 提示模板是一个 “未完成” 的输入示例。Langchain 提供了我们通过告诉它要替换占位符的值来创建格式化之后的提示。Langchain 内置的提示模板对象 (PromptTemplate), 使我们能够将用户输入插入到提示文本的占位符中，并且可以增加额外的知识,额外的上下文, 将用户输入进行格式化，最终提供给语言模型高质量的提示,从而获得更好的输出结果。

#### 提示模板的定义

在Langchain中，提示模板是一种可复制生成提示的方式。它包含一个文本字符串（即”模板”），该模板可以接收用户的一系列参数，从而生成一个提示。比如说，我们要给公司的产品取一个好听的名字，用户输入的就是产品的品类名字，例如“袜子”或“毛巾”。然而，我们并不需要为每一个品类都编写一个提示，而是使用这个模板，根据用户输入的不同品类，生成对应的提示。

#### 模板的三个组成部分

提示模板主要由三个部分组成：语言模型的说明，一组少数示例，以及一个问题或任务。这三部分中，语言模型的说明和示例是固定的常量，而问题或任务是可以由用户改变的变量。Langchain将这三个部分组合起来，并为输出结果进行格式化的处理，以生成一个完整的提示。

提示模板在Langchain中得到了广泛的应用，但也在许多其他的大语言模型提示系统中使用。我们常常能看到类似的情况。Langchain的提示模板在其他模型中是通用的。

提示模板有许多实际应用场景。比如，它在做摘要或分类时非常实用。以新闻摘要为例，我们可以构建一个模板，模型的任务就是生成新闻的摘要。用户的输入可能是复制黏贴了一篇完整的新闻文章，给语言模型的说明可能是“生成这篇新闻的摘要”，示例则可以是一些已经生成过的新闻摘要。然后，Langchain将这三个部分组合起来，生成一个完整的提示，如：“请为这篇新闻生成一个摘要。” 根据这个提示，模型就能生成一个符合要求的新闻摘要。这样，我们就可以利用提示模板，轻松地为任意一篇新闻生成摘要。

#### 一个最简单的示例

这是最基本的示例,这个提示模板没有包含示例,但它使我们能够动态更改带有用户输入的提示。

from langchain import PromptTemplate  
  
template = """  
You are an expert data scientist with an expertise in building deep learning models.   
Explain the concept of {concept} in a couple of lines  
"""  
  
prompt = PromptTemplate(  
 input\_variables=["concept"],  
 template=template,  
)  
  
prompt.format(concept="NLP")

当我们传入 concept="NLP" 时, PromptTemplate 方法,将 “NLP” 注入到模板中

我们得到的结果是:

'\nYou are an expert data scientist with an expertise in building deep learning models. \nExplain the concept of NLP in a couple of lines\n'

这个代码示例将用户输入变量 input\_variables和开发者定义的 template 传参给 PromptTemplate对象, 然后调用 format 方法，格式化为语言模型可用的字符串提示。

#### 构造提示模板的步骤

构造模板包含取出模板和格式化两个步骤。from\_template即取出模板，format即格式化用户输入。Langchain, 内置了很多模板，比如聊天模型的提示模板，角色的提示模板。格式化可以是普通字符串用作通用语言模型输入，也可以是用作聊天模型输入的Message对象。

上面这个代码还可以简化为：

# 导入PromptTemplate和template定义同上  
  
prompt = PromptTemplate.from\_template(template).format(concept="NLP")  
prompt

这段代码可以用自然语言描述为：取出内置模式给template使用，传参用户的输入后格式化输出完整提示。无论是哪一种提示模板，都是基于 PromptTemplate 对象，我们基于这个对象构造自己的提示模板。

内置提示模板，是Langchain简化开发的方式。即使你不指定 input\_variables , Langchain 会自行推断，只要你告诉了它取出哪个模式。

#### 专用聊天模型的提示模板

专用聊天模型的提示模板的不同之处，输入的是消息列表, 支持输出 Message 对象。Langchain 的优势在于提供了聊天提示模板 （ChatPromptTemplate）和角色消息提示模板。角色消息提示模板包含AIMessagePromptTemplate, SystemMessagePromptTemplate 和 HumanMessagePromptTemplate 三种角色消息提示模板。

无论看起来这些多么复杂，他们都遵循构造模板的步骤，取模式和格式化。聊天模型的提示模板取是消息提示模式，格式化多了消息列表的选择。

我们将上一个示例代码改造为聊天提示模板，下面是代码示例：

先导入消息提示模板：

from langchain.prompts import (  
 ChatPromptTemplate,  
 PromptTemplate,  
 SystemMessagePromptTemplate,  
 AIMessagePromptTemplate,  
 HumanMessagePromptTemplate,  
)

改造思路是生成人类和系统的消息提示，通过ChatPromptTemplate整合为消息列表提示： 先使用 from\_template 取出消息模式，传入定义的template模板字符串，生成人类和系统两种模板。

from langchain import PromptTemplate  
  
template = """  
You are an expert data scientist with an expertise in building deep learning models.   
"""  
  
system\_message\_prompt = SystemMessagePromptTemplate.from\_template(template)  
human\_template="Explain the concept of {concept} in a couple of lines"  
human\_message\_prompt = HumanMessagePromptTemplate.from\_template(human\_template)

然后转化为聊天提示模板类型：

chat\_prompt = ChatPromptTemplate.from\_messages(system\_message\_prompt, human\_message\_prompt])

最后指定输出为消息类型的格式, 字符串的输出则是这样：

chat\_prompt.format(concept="NLP")

打印输出的是：

'System: \nYou are an expert data scientist with an expertise in building deep learning models. \n\nHuman: Explain the concept of NLP in a couple of lines'

指定格式化为消息对象类型则是这样：

chat\_prompt.format\_prompt(concept="NLP").to\_messages()

打印输出的是：

[SystemMessage(content='\nYou are an expert data scientist with an expertise in building deep learning models. \n', additional\_kwargs={}),  
HumanMessage(content='Explain the concept of 强化学习 in a couple of lines', additional\_kwargs={}, example=False)]

尽管上述示例并没有展示如何加入示例，但请不要忧虑，接下来我们即将进入下一章节：少样本提示模板。这个章节将全面展示如何利用示例来优化我们的模板。

事实上，所有的提示模板都基于示例选择这一概念。为了从大语言模型中获取更加完整和准确的结果，少量示例学习（Few Shot Learning）就显得尤为重要。下一章少样本提示模板，就是在构造提示时，我们提供一些示例来指导模型的输出。这些示例可以帮助模型更好地理解任务要求，从而生成更加贴合用户预期的答案。这也正是FewShotPromptTemplate的价值所在：通过使用少量示例学习，我们能够有效提升模型的输出质量，让模型给出的回答更加准确和具有针对性。