# 2.3.1 输出列表格式

数组或者列表是程序世界最基本的数据格式, 这种格式在大语言模型开发中,也可以发挥不少作用.

你可以想象大语言模型就像一个非常聪明的机器人，它可以回答很多问题。但是，有时候这个机器人会说一些很复杂或者不正确的话，就像它在猜你想要知道什么，或者想要显得自己很聪明一样。

我们可以让这个机器人把它的回答写在一张“清单”上，就像你写购物清单一样。这样，我们就可以很容易地看到它的每个答案，而不需要去理解一大堆复杂的话。这个“清单”还可以帮助我们更好地使用这个机器人的答案，比如我们可以把这个名单给到其他机器人让它们帮忙处理。

使用“清单”这样的方式，可以让我们更好地使用这个聪明的机器人，让它变得更有用，更容易理解。

在软件世界中,最常见的“清单”就是列表格式。

我们通过快速入门的代码,来看看 Langchain 是如何将模型的回答输出为“清单”式的列表格式。

下面是最简单的列表输出解析器代码：

导入库,实例化列表输出解析器对象（CommaSeparatedListOutputParser）, 预期我们获得列表的结果:

from langchain.output\_parsers import CommaSeparatedListOutputParser  
from langchain.prompts import PromptTemplate, ChatPromptTemplate, HumanMessagePromptTemplate  
from langchain.llms import OpenAI  
from langchain.chat\_models import ChatOpenAI  
  
output\_parser = CommaSeparatedListOutputParser()

获取格式化指令:

format\_instructions = output\_parser.get\_format\_instructions()  
prompt = PromptTemplate(  
 template="List five {subject}.\n{format\_instructions}",  
 input\_variables=["subject"],  
 partial\_variables={"format\_instructions": format\_instructions}  
)

这里我们引入 OpenAI 模型:

model = OpenAI(temperature=0)

实例化一个提示模板:

\_input = prompt.format(subject="ice cream flavors")  
output = model(\_input)

将提示传入OpenAI 模型:

\_input = prompt.format(subject="ice cream flavors")  
output = model(\_input)

调用解析器的parse方法,解析数据为列表格式.

output\_parser.parse(output)

最终的结果是:

['Vanilla',  
 'Chocolate',  
 'Strawberry',  
 'Mint Chocolate Chip',  
 'Cookies and Cream']

从快速入门代码,我们使用了输出解析器的2大方法: 格式化 output\_parser.get\_format\_instructions() 和 解析 output\_parser.parse()

接下来,我们进入到语言模型应用开发中，最常用的一种格式： JSON 对象。