

【Day 4】LangChain (2)

Runnables 介紹

1. LangChain Runnables

LangChain 的核心優勢在於它的**模組化與可組合性**。本篇將介紹重要的 `Runnable`，讓你能夠將客製化的函數與邏輯整合到流程中。熟練掌握它們能讓你兼顧 LangChain 的強大功能，同時避免過度封裝的缺點。以下的例子改編自 langchain 的 documentation。

1.1 RunnableLambda

將 python 的 callable（像是函數）轉換為 LangChain 的 `Runnable`，如此函數就能被 `invoke` 或 `stream` 等。然而，當需要對串流輸出的 `chunks` 進行處理或是進行串流輸出，我們要用 `RunnableGenerator` 來把 python 的 generator 變成 `Runnable`。

```
from langchain_core.runnables import RunnableLambda, RunnableParallel,
RunnableSequence, RunnablePassthrough

def mul_two(x: int) → int:
    return x * 2

def mul_three(x: int) → int:
    return x * 3

runnable_mul_two = RunnableLambda(mul_two)
runnable_mul_three = RunnableLambda(mul_three)
```

1.2 RunnableSequence

`RunnableSequence` 是線性的流程：前一個 `Runnable` 的輸出就是後一個 `Runnable` 的輸入。當左或右其一是 `Runnable` 時可以用 `|` 運算符直接定義，例如最常見的用法如：`prompt | llm`。

```
# runnable_1 | runnable_2
RunnableSequence(
    runnable_mul_two,
    runnable_mul_three
).invoke(1) # 6
```

1.3 RunnableParallel

`RunnableParallel` 會並行式（concurrently）的運行每一個 key，並回傳一個字典。可以在 `sequence` 裡面用 `{}` 直接定義。

```
# return a dict. run the runnables concurrently with the same input
RunnableParallel(
    mul_2=runnable_mul_two,
    mul_3=runnable_mul_three
).invoke(1) # {'mul_2': 2, 'mul_3': 3}
```

1.4 Runnable Passthrough

`RunnableParallel` 是 `runnable` 系列中的 `identity`，不會對輸入進行任何動作。

```
RunnableParallel(
    origin=RunnablePassthrough(),
    modified=runnable_mul_two
).invoke(1) # {'origin': 1, 'modified': 2}
```

使用類（class）方法 `RunnablePassthrough.assign()` 時，會在輸入的內容之上，再加入一個新的 key；如果傳入的字典已有指定的新的 key，會覆蓋它。

```
RunnablePassthrough().assign(
    c=lambda x: x['a'] + x['b'],
).invoke({'a': 1, 'b': 1, 'c': 1}) # {'a': 1, 'b': 1, 'c': 2}
```

1.5 其他 Runnables

除了上述最常見的幾個 `runnables` 之外，我還會用文字描述以下幾個 `runnables` 的特性。

RunnableWithFallbacks

這是帶有 fallbacks 的 runnables，讓我們可以定義原本的流程出錯的時候，要改為執行什麼 runnable。可以用 runnables 的 `.with_fallbacks()` 方法來定義：

Runnable Assign

類似於 `RunnablePassthrough.assign()` 的功能，也能新增 key，差別是在建立的時候要傳入一個 `RunnableParallel`。輸出會將輸入以及該 `RunnableParallel` 的輸出合併在一起。

RunnablePick

在建立時使用 `keys` 參數（傳入一個列表）來選定需要保留的 key。

Runnable Branch

條件式的執行給定的 runnables，傳入參數為 (Condition, Runnable) 組成的 list。列表的最後是所有 conditions 不成立時執行的 runnable。

Runnable Binding

通常使用 runnables 的 `.bind()` 方法來定義，例如 `llm.bind(temperature=0.2)`。它會回傳另一個綁定了指定參數的 runnable。在概念上，這與 `functools.partial` 有些相似。

Runnable Each

`RunnableEach` 包裝過後的 runnable，會在輸入列表的每一個元素上套用該 runnable，並返回一個結果的列表。

Runnable{X} 記憶圖表

Runnable{X}



