04.Planner.md 2024-01-01

第四章: Planner 组件 - 让大模型有规划地工作

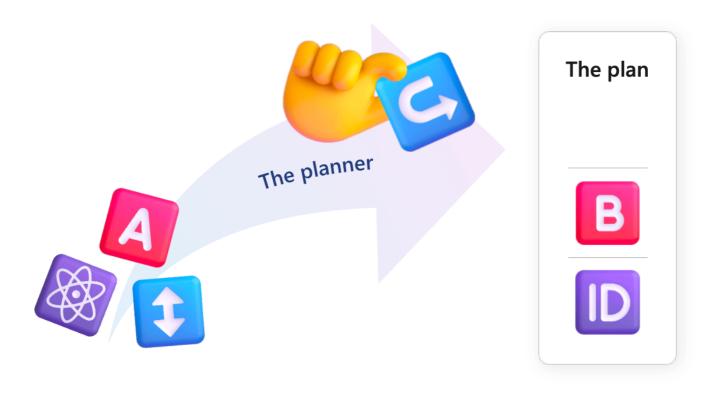
我们在第三章学习了 Semantic Kernel 非常重要的功能-插件,通过插件可以完成针对不同领域的工作。 LLMs 改变了人机交互的方式,通过自然语言去与大模型对话,让大模型完成工作。但往往我们给出的指令不只是完成单一的工作,如"请根据西雅图的天气给出差的人群发一个穿衣提醒的邮件"。我们一直希望人工智能和人类作对比,那我们把上述指令用人的思维去思考,就会有如下拆分:

- 1. 查询西雅图天气
- 2. 从公司系统查询出差的人以及其联系方式
- 3. 穿衣提醒邮件模版
- 4. 发送邮件

LLMs 其实都有同样的思维,在 Semantic Kernel 中就有强大的 Planner 功能来做任务拆分的事情。本章就会和大家讲述相关的内容。

什么是 Planner

Planner 是 Semantic Kernel 一个重要组件,通过它可以接收任务指令,然后与已经在 Kernel 定义好的内置插件或者自定义插件做对应,让任务指令按部就班来工作。 就如开篇所提及的内容,实际上针对 "请根据西雅图的天气给出差的人群发一个穿衣提醒的邮件" 的指令,会现在 plugins 定义相关的插件,并通过 Kernel 注册后,Semantic Kernel 就协助您去配对步骤。



如何使用 Planner

现在 Planner 还在进化阶段,在最新的 .NET 版本中,我们发现 Semantic Kernel 关于 Planner 组件 Microsoft.SemanticKernel.Planners.Handlebars 的版本与核心的 Microsoft.SemanticKernel 有所不同。有用 04.Planner.md 2024-01-01

户在质疑是否 Semantic Kernel 的版本号混乱。你可以理解在 2023 年 Semantic Kernel 团队完成了核心部分,至于针对组件化的功能如 Planner ,如 Memory ,如部分 Connector 等都还在进化。毕竟这些功能和 LLMs 的发展有关。

如果您需要使用 Planner,需要考虑您的业务场景。例如在一些业务流程中加入 Planner ,以及工具链中加入 Planner 都是非常有用的,毕竟人类对工作自动化有很多的思考

.NET 场景下

需要添加关于 Planner 的相关组件库

```
#r "nuget: Microsoft.SemanticKernel.Planners.Handlebars, *-*"
```

引用库

```
using Microsoft.SemanticKernel.Planning;
```

注意:在采用 HanlerBars 使用时,你需要注意注意这些功能还在演变,在使用时请忽略 SKEXP0060

#pragma warning disable SKEXP0060

Python 场景下

直接引用库

from semantic_kernel.planning.basic_planner import BasicPlanner

或

from semantic_kernel.planning.sequential_planner import SequentialPlanner

04.Planner.md 2024-01-01

注意: 这里 Python 的 Planner 和 .NET 的 Planner 设定有所不同, Python 应该和 .NET 同步,所以大家在 Python 使用时,需要有以后变更的准备

改变中的 Planner

在官方的博客中,有所提及 Planner 的变化 https://devblogs.microsoft.com/semantic-kernel/migrating-from-the-sequential-and-stepwise-planners-to-the-new-handlebars-and-stepwise-planner/ 结合了 Function Calling 的调用重新整理了在预览版中不同的 Planner 整合,大家可以关注一下该内容来了解。

如果你希望了解 Planner 实现的原理,请参考

https://github.com/microsoft/semantic-

kernel/blob/main/dotnet/src/Planners/Planners.Handlebars/Handlebars/CreatePlanPrompt.handlebars

例子

.NET 例子请点击访问这里

Python 例子请点击访问这里

小结

Planner 的加入让 Semantic Kernel 的可用性大大提高,特别针对业务和工具类的场景。构建企业级的插件库,对于 Planner 落地也是非常重要,毕竟我们通过插件组合出不同的任务来完成工作。