

Agent memory

随着Agent承担更多涉及众多用户互动的复杂任务，为它们配备记忆同样对效率和用户满意度至关重要。有了记忆，Agent可以从反馈中学习，并适应用户的偏好。

Agent记忆机制示例

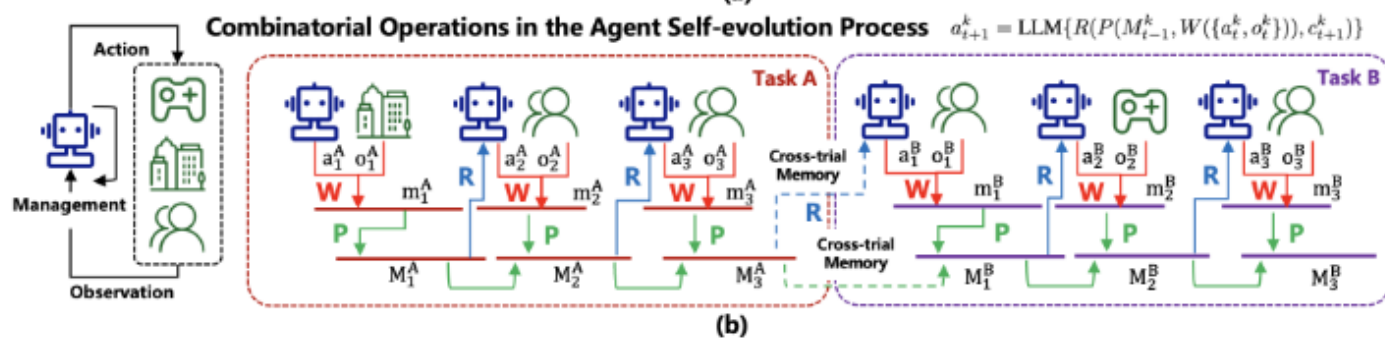
Task A: Making a trip plan for Alice in Beijing



Task B: Recommending a movie for Alice



(a)



(b)

(a) Agent与环境交互过程中潜在试验的示例。(b) 记忆的读取、写入和管理过程的说明，其中虚线表示跨试验信息可以被整合到记忆模块中。

在上图例子中，Agent有两个任务：

- (A) 为Alice在2024年5月1日至5月3日期间在北京制定旅行计划，Agent需要(i) 购买往返机票，(ii) 根据Alice的偏好确定几个景点，(iii) 安排这些景点的参观顺序。
- (B) 在2024年5月10日为Alice推荐一部电影，Agent需要(i) 确定Alice有空看电影的时间，(ii) 根据Alice的观看偏好选择一部电影。对于每个任务，在下面展示一个可能的试验：

对于任务(A)，试验是：

- [步骤 1] Agent：我需要通过在线票务办公室预订2024年5月1日从上海到北京的航班，以及5月3日从北京返回上海的航班。环境（在线票务办公室）：订单已确认。出发航班上午9:00起飞，您的票号是EK240501SX9021。返程航班下午4:30起飞，您的票号是LG240503FN6317。请准时到达机场搭乘您的航班。
- [步骤 2] Agent：我已经从《北京景点》杂志获取了外部知识。它显示长城、故宫和颐和园都很有名。根据爱丽丝的偏好，我认为她更喜欢古代建筑，所以我确定故宫和颐和园作为目的地。环境（Alice）：做得好！
- [步骤 3] Agent：现在，我需要为Alice安排参观顺序。根据上面的信息，我建议她一到北京就参观颐和园，因为那里适合整个下午游玩并观看日落。然后，她第二天可以去故宫，这样会给她足够的时间。
- 最后，她可以在第三天返回上海休息。环境（Alice）：我喜欢这个计划。谢谢！

对于任务(B)，试验是：

- [步骤 1] Agent：根据我以前为Alice推荐电影的经验，她更喜欢在晚上9:00开始看电影，因为那时她已经完成工作。我应该问问Alice那个时间她是否有空。环境（Alice）：当然，这是个好时间！
- [步骤 2] 智能体：现在，我需要找到一些Alice在Netflix上标记为“想看”的电影。环境（Netflix）：Alice的想看列表：《星际穿越》、《恐怖星球》和《鬼吹灯》。
- [步骤 3] Agent：由于Alice晚上从不观看恐怖电影，《星际穿越》对她来说更合适。现在，我需要与爱丽丝确认这部电影。环境（Alice）：太棒了！我喜欢它！

实操demo

Mem0是一个为AI应用设计的智能记忆层，旨在通过保留用户偏好并随时间不断适应，提供个性化且高效的交互体验。特别适合聊天机器人和AI驱动的工具，Mem0能够创建无缝、上下文感知的体验。

下面将通过介绍Mem0记忆管理的基本操作来做一个Agent持有记忆的简单demo，同时使用Milvus，一个高性能、开源的向量数据库，它能够支持高效的存储和检索。这个实践入门指南将引导你完成基础记忆操作，帮助你使用Mem0和Milvus构建个性化的AI交互。

```
1 ! pip install mem0ai pymilvus
```

配置Mem0与Milvus

```
1 import os
2 os.environ["OPENAI_API_KEY"] = "sk-*****"
```

现在，我们可以配置Mem0使用Milvus作为向量存储。

```
1 # Define Config
2 from mem0 import Memory
3
4 config = {
5     "vector_store": {
6         "provider": "milvus",
7         "config": {
8             "collection_name": "quickstart_mem0_with_milvus",
9             "embedding_model_dims": "1536",
10            "url": "./milvus.db", # Use local vector database for demo purpose
11        },
12    },
13    "version": "v1.1",
14 }
15
16 m = Memory.from_config(config)
```

使用Mem0和Milvus管理用户记忆库

添加记忆

add函数将非结构化文本作为记忆存储在Milvus中，并将其与特定用户和可选元数据关联。

在这里，将Alice的记忆“working on improving my tennis skills”连同相关metadata一起添加到Milvus中。

```
1 # Add a memory to user: Working on improving tennis skills
2 res = m.add(
3     messages="I am working on improving my tennis skills.",
4     user_id="alice",
5     metadata={"category": "hobbies"},
6 )
7
8 res
```

输出结果：



```
{'results': [{'id': '77162018-663b-4dfa-88b1-4f029d6136ab',
  'memory': 'Working on improving tennis skills',
```

```
'event': 'ADD']],  
'relations': []}
```

搜索记忆

可以使用搜索功能来寻找与用户最相关的记忆。

让我们从为Alice添加另一个记忆开始。

```
1 new_mem = m.add(  
2     "I have a linear algebra midterm exam on November 20",  
3     user_id="alice",  
4     metadata={"category": "task"},  
5 )
```

现在，调用 `get_all` 函数并指定 `user_id` 来验证我们确实为用户alice保存了2条记忆记录。

```
1 m.get_all(user_id="alice")
```

输出结果：

```
🎵 {'results': [{'id': '77162018-663b-4dfa-88b1-4f029d6136ab',  
  'memory': 'Likes to play tennis on weekends',  
  'hash': '4c3bc9f87b78418f19df6407bc86e006',  
  'metadata': None,  
  'created_at': '2024-11-01T19:33:44.116920-07:00',  
  'updated_at': '2024-11-01T19:33:47.619857-07:00',  
  'user_id': 'alice'},  
  {'id': 'aa8eaa38-74d6-4b58-8207-b881d6d93d02',  
  'memory': 'Has a linear algebra midterm exam on November 20',  
  'hash': '575182f46965111ca0a8279c44920ea2',  
  'metadata': {'category': 'task'},  
  'created_at': '2024-11-01T19:33:57.271657-07:00',  
  'updated_at': None}]}
```

```
'user_id': 'alice']}]}
```

可以进行搜索，通过提供查询内容和用户ID来寻找与用户最相关的记忆。默认情况下，使用L2度量（欧几里得距离）来进行相似度搜索，因此，得分越小意味着相似度越高。

```
1 m.search(query="What are Alice's hobbies", user_id="alice")
```

输出结果：



```
{'results': [{'id': '77162018-663b-4dfa-88b1-4f029d6136ab',  
  'memory': 'Likes to play tennis on weekends',  
  'hash': '4c3bc9f87b78418f19df6407bc86e006',  
  'metadata': None,  
  'score': 1.2807445526123047,  
  'created_at': '2024-11-01T19:33:44.116920-07:00',  
  'updated_at': '2024-11-01T19:33:47.619857-07:00',  
  'user_id': 'alice'},  
 {'id': 'aa8eaa38-74d6-4b58-8207-b881d6d93d02',  
  'memory': 'Has a linear algebra midterm exam on November 20',  
  'hash': '575182f46965111ca0a8279c44920ea2',  
  'metadata': {'category': 'task'},  
  'score': 1.728922724723816,  
  'created_at': '2024-11-01T19:33:57.271657-07:00',  
  'updated_at': None,  
  'user_id': 'alice'}]}
```