一、具有语义和情景记忆的邮件助手 *00:01*
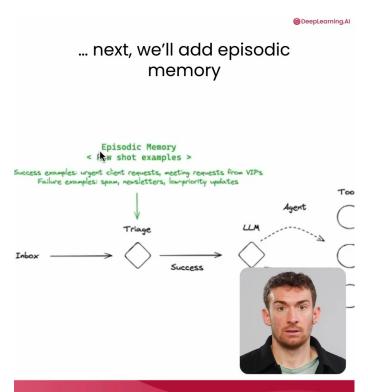
1. 上节回顾 00:03

-
- **语义记忆**: 在上一课中已添加到代理中，存储关于用户的事实性信息，如"Things I learned in school"对应人类记忆中的事实知识。
- **情景记忆引入**: 本节课将添加情景记忆，以few shot示例的形式集成到分类步骤中。

2. 定义长期记忆存储 00:51

Memory Type	What is Stored	Human Example	Agent Example
Semantic	Facts	Things I learned in school	Facts about a user
Episodic	Experiences	Things I did	Past agent actions
Procedural	Instructions	Instincts or motor skills	Agent system prompt

-

- **存储结构**: 使用create_manage_memory_tool和create_search_memory_tool创建记忆工具，命名空间格式为 ("email_assistant","{langgraph_user_id}","collection")
- **情景记忆本质**: 代表代理的经验，通常以过去的代理操作和few shot示例形式存在，这些示例将被传递到提示中。

3. 定义few shot示例 01:06

```python
from langgraph.store.memory import InMemoryStore

store = InMemoryStore(
    index={"embed": "openai:text-embedding-3-small"}
)
```

```
/opt/conda/lib/python3.11/site-packages/langgraph/store/base/embed.p
y:95: LangChainBetaWarning: The function `init_embeddings` is in bet
a. It is actively being worked on, so the API may change.
  return init_embeddings(embed)
```

-
- **记忆类型对比**:
  - **语义记忆**: 存储事实，如"Facts about a user"
  - **情景记忆**: 存储经验，如"Past agent actions"
  - **程序记忆**: 存储指令，如"Agent system prompt"
- **示例格式**:

```
email = {
    "author": "Alice Smith <alice.smith@company.com>",
    "to": "John Doe <john.doe@company.com>",
    "subject": "Quick question about API documentation",
    "email_thread": """Hi John,

I was reviewing the API documentation for the new authentication serv

Specifically, I'm looking at:
- /auth/refresh
- /auth/validate

Thanks!
Alice""",
}
```

```
data = {                                    ⧉  ↑  ↓  ⊥  ⊽  🗑
    "email": email,
    # This is to start changing the behavior of the agent
    "label": "respond"
}
```

-
- **存储方式**: 将示例放入长期记忆存储，使用不同的命名空间区分语义记忆("collection")和情景记忆("examples")

4. 定义三重记忆邮件助手 04:28

```
print(format_few_shot_examples(results))

Here are some previous examples:

------------

Email Subject: Update: Backend API Changes Deployed to Staging
Email From: Sarah Chen <sarah.chen@company.com>
Email To: John Doe <john.doe@company.com>
Email Content:
```
Hi John,

    Just wanted to let you know that I've deployed the new authentic
ation endpoints we discussed to the staging environment. Key changes
include:

    - Implemented JWT refresh token rotation
    - Added rate limiting for login attempts
    - Updated API documentation with new endpoints

    All tests are passing and the changes are ready for review. You
can test it out at st
```
> Triage Result: ignore
```

-
- **分类系统提示**:
  ○ 角色: 执行助理
  ○ 背景: 包含用户个人资料
  ○ 指令: 将邮件分为IGNORE/NOTIFY/RESPOND三类

- 规则: 定义每类邮件的具体标准
- Few shot示例: 展示历史分类案例

```
email_agent = StateGraph(State)
email_agent = email_agent.add_node(triage_router)
email_agent = email_agent.add_node("response_agent", response_agent)
email_agent = email_agent.add_edge(START, "triage_router")
email_agent = email_agent.compile(store=store)
}
```

```
response = email_agent.invoke(
    {"email_input": email_input},
    config={"configurable": {"langgraph_user_id": "harrison"}}
)
```

```
Classification: RESPOND — This email requires a response
```

- **路由节点增强**:
  - 添加配置和存储参数
  - 在格式化系统提示前添加few shot示例搜索逻辑
  - 使用format_few_shot_examples函数将检索结果格式化为字符串

5. 应用案例 07:22

1) 例题:邮件响应分类

```
data = {
    "email": {
    "author": "Tom Jones <tome.jones@bar.com>",
    "to": "John Doe <john.doe@company.com>",
    "subject": "Quick question about API documentation",
    "email_thread": """Hi John - want to buy documentation?""",
},
    "label": "ignore"
}
```

```
store.put(
    ("email_assistant", "harrison", "examples"),
    str(uuid.uuid4()),
    data
)
```

- **初始行为**: 对于"Hi John - want to buy documentation?"邮件，默认分类为RESPOND
- **行为修改**:
  - 添加标记为"ignore"的few shot示例
  - 示例存储在("email_assistant", "harrison", "examples")命名空间
  - 修改后相同邮件被分类为IGNORE

```python
        return [
            {
                "role": "system",
                "content": agent_system_prompt_memory.format(
                    instructions=prompt_instructions["agent_instructions"]
                    **profile
                )
            }
        ] + state['messages']
```

```python
from langgraph.prebuilt import create_react_agent
```

```python
tools= [
    write_email,
    schedule_meeting,
    check_calendar_availability,
    manage_memory_tool,
    search_memory_tool
]
response_agent = create_react_agent(
    "anthropic:claude-3-5-sonnet-latest",
    tools=tools,
    prompt=create_prompt,
    # Use this to ensure the store is passed to the agent
    store=store
)
```

```python
config = {"configurable": {"langgraph_user_id": "lance"}}
```

- **泛化能力**: 即使修改邮件内容(如添加更多问号或收件人)，分类结果保持一致
- **用户隔离**: 不同用户ID(langgraph_user_id)保持独立的分类行为

二、知识小结

知识点	核心内容	考试重点/易混淆点	难度系数
情景记忆（Episodic Memory）的集成	通过少量示例（few-shot examples）形式添加到智能体，并整合到分类（triage）步骤中	情景记忆与语义记忆的区别（**经验 vs. 事实**）	★★
长期记忆存储（Long-term Memory Store）	定义命名空间（如 email_assistant_\<userID\>_examples），存储输入（邮件内容）和输出（标签/响应）	命名空间与语义搜索的collection区别	★★
少量示例的格式化与检索	通过相似性搜索返回匹配示例，并转换为易读字符串（含邮件主题、发件人、分类结果）	**非精确匹配**（语义相似性优先）	★★★

动态分类逻辑调整	通过用户添加的示例（如标记ignore的邮件）实时更新分类行为	用户ID作用域（不同用户示例独立）	★★★★
智能体测试与验证	修改邮件内容（如添加问号或收件人）验证分类一致性，切换用户ID测试作用域隔离	示例泛化能力（相似邮件处理一致性）	★★★

- 5-具有语义+情景记忆的邮件助手 — 具有语义和情景记忆的邮件助手
  - 上节回顾 ⭐
    - 上节内容回顾
    - 本节课内容概述
    - 情景记忆添加形式
    - 情景记忆添加位置
    - 情景记忆定义
    - 情景记忆作用
    - 演示准备
    - 接下来步骤
  - 定义长期记忆存储
    - 定义长期记忆存储
    - 数据加载
  - 定义few shot示例 ⭐⭐⭐
    - Few shot示例定义
    - Few shot示例存储
    - Few shot示例使用
    - Helper函数作用
    - 示例格式化内容
    - Few shot搜索模拟
    - Few shot搜索应用
  - 定义三重记忆邮件助手 ⭐⭐
    - 定义三重记忆邮件助手
    - 步骤一
    - 步骤二
    - 步骤三
    - 步骤四
    - 步骤五
    - 步骤六
  - 应用案例 — 例题:邮件响应分类
    - 邮件响应分类例子
    - 默认响应
    - 改变响应方式
    - 存储位置
    - 效果展示
    - 用户ID的作用
    - 新增功能
    - 课程结尾