人工智能：Agent

饶洋辉
计算机学院，
中山大学
raoyangh@mail.sysu.edu.cn
http://cse.sysu.edu.cn/node/2471

课件来源：中山大学余超教授等

目录

- 概述
- Agent及其结构
- 协调与协作
- 基于大语言模型的Agent

概述

Agent（智能体）的概念与示例

- 智能体的概念应用广泛，在不同的研究领域对它有不同的理解。
- 在人工智能领域，智能体是指驻留在环境下能够自主、灵活地实施行为以满足设计目标的行为实体。
- 具体的，智能体具有几个方面的特征：
  - **智能体是行为实体**
  - **智能体驻留在环境中（驻留性）**
  - **智能体行为具有自主性**
  - **智能体行为具有不同程度的灵活性**

概述

智能体是行为实体

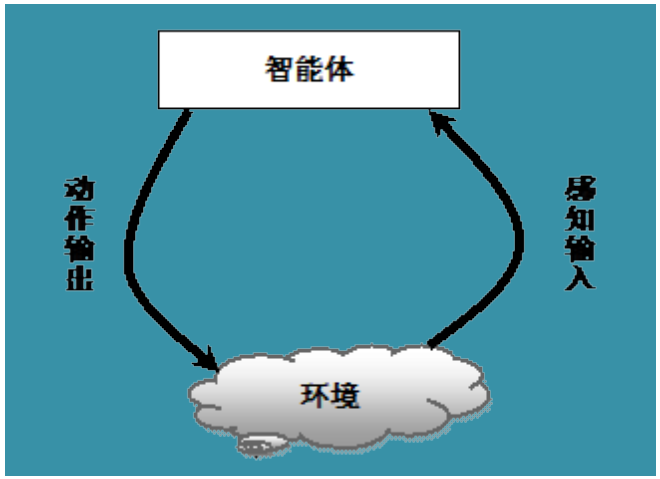
- 智能体是具有行为能力的实体，它具有一组动作并能执行这些动作，是动作执行的拥有者和实施者。智能体的动作反映了智能体所具有的能力，这种能力主要体现为两个不同的方面。
  - 从内部角度上看，智能体动作的实施将有助于实现智能体的任务和目标；
  - 从外在行为上看，智能体的动作实施将对智能体所驻留的环境产生影响。
- 一般地，不具备行为实施能力的实体通常不被视为是智能体。如：数据、资源、信息等。
- 负责对数据、资源、信息等进行管理、维护的实体可以被视为是智能体。例如，服务机器人是一个智能体，它具有行走、转弯、抓取物体等行为。

概述

智能体驻留在环境中（驻留性）

- 任何智能体都不是孤立和封闭的，它驻留在一定的环境之中，需要与环境进行交互，获取环境及其变化信息，以此来指导其行为实施。

- 如图所示，智能体与其驻留环境之间的交互主要表现为两个方面：



  - 通过感知输入获得环境变化
  - 通过动作输出来影响环境。

概述

智能体行为具有自主性

- 智能体能在没有人类或其他智能体的干涉和指导的情况下运行，并能根据其内部状态和感知到的环境信息来决定该实施什么样的动作，进而控制自身的行为。

- 简单的讲，智能体行为的自主性是指智能体在适当的场景下知道该选择什么样的动作来执行。

概述

智能体行为具有不同程度的灵活性

- 智能体行为的灵活性主要体现为以下三方面：
  - 反应性（Reactive）
  - 自发性（Pro-active）
  - 社会性（Social）

- 反应性是指智能体能够感知所处的环境，并能对环境中发生的相关事件（比如智能体间的交互、用户的指令等）作出适时反应，以满足系统设计目标。

例如：家庭服务机器人一旦遇到障碍物，它需要立即实施躲避行为，以免发生碰撞。

概述

- 自发性是指智能体不仅具有目标并根据目标行事，而且能够主动地产生目标，进而实施自发的行为。

例如：一旦发现有陌生人访问家庭，服务机器人会自发产生一个目标：将来访者信息告知用户，进而主动实施一组行为，如呼叫用户的手机、给用户发送短信等。

- 社会性是指智能体所驻留的环境可能存在其他智能体，它拥有这些智能体的有关信息和知识（比如这些智能体的物理位置、所拥有的资源和能力等），并能与它们进行交互和通信，实现灵活多样和复杂的合作、协商和竞争，以满足系统的设计目标。

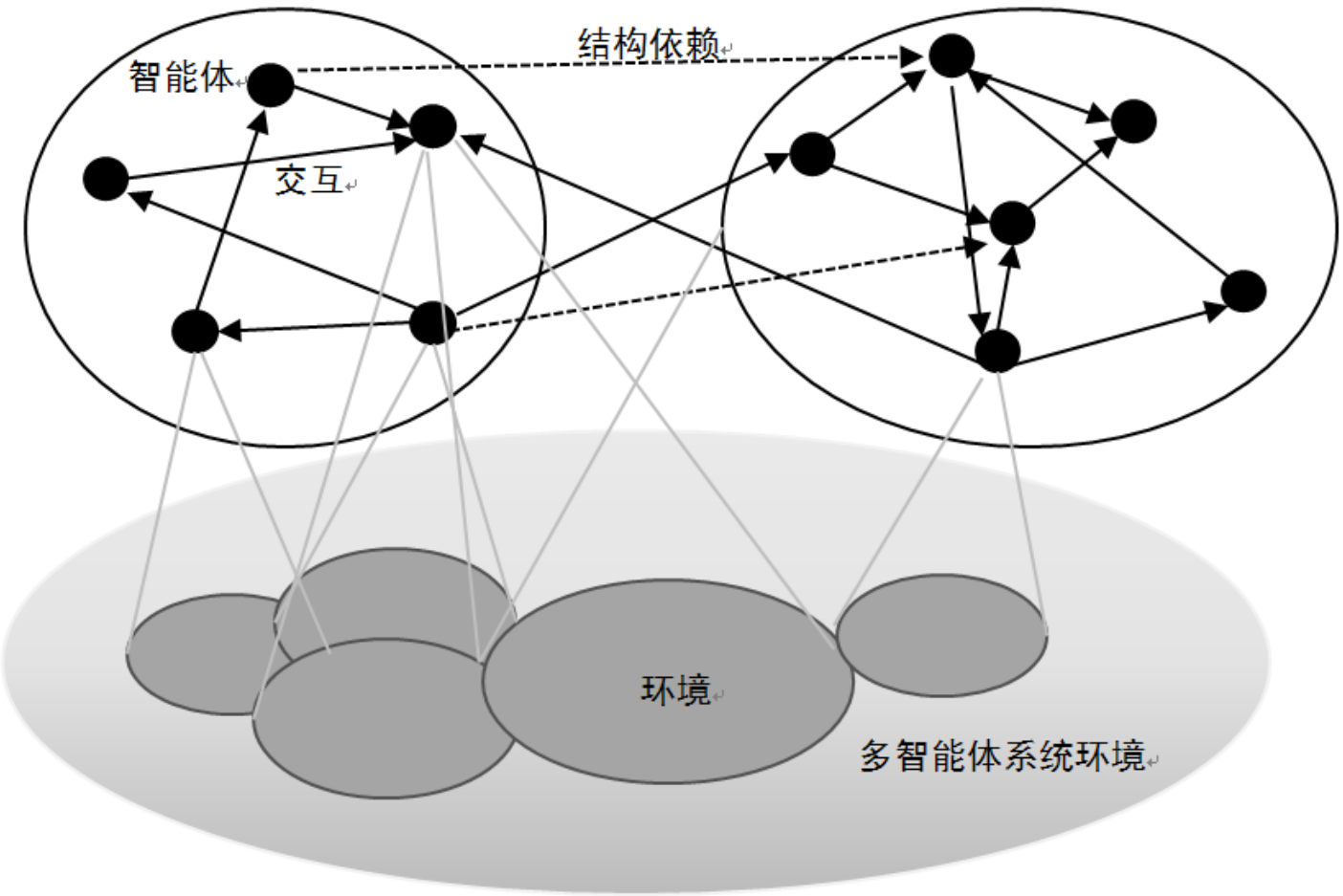
例如：家庭服务机器人能够与互联网上的气象信息提供智能体进行交互，从而获取天气信息。

概述

多智能体系统的概念与示例

- 多智能体系统是指由多个相对独立同时又相互作用的智能体所构成的系统。
  - 独立性：每个智能体都是自主的行为实体，封装了行为以及行为控制机制，可以在无需外部指导的情况下实施行为；
  - 协同性：这些智能体并不是孤立的，它们之间存在各种关系，需要相互交互和协同进而达成问题的求解。

概述



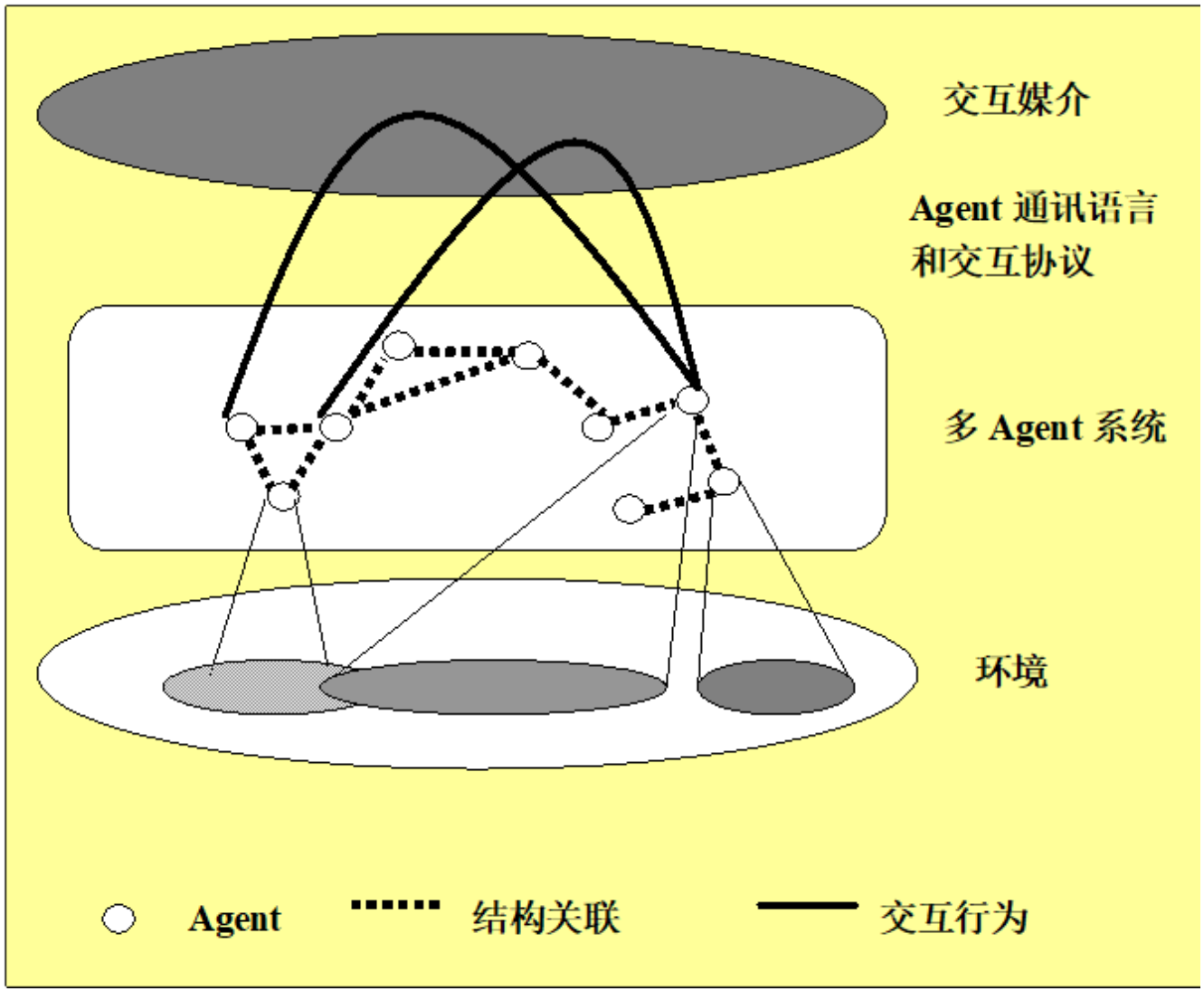
图：多智能体系统的高层视角

概述

多智能体系统的特性

- **无全局视角**：对于绝大多数多智能体系统而言，系统中的智能体要获得关于系统以及环境的完整、准确和及时的信息是非常困难的。这是由于智能体所驻留的环境可能会比较复杂，具有以下特点：
  - 开放性
  - 不确定性
  - 动态性等特点。
- **不可预测和不确定性**：在多智能体系统中，由于智能体行为的自主性、智能体行为对环境影响的有限性、系统中无全局控制智能体、智能体间交互行为的动态性，多智能体系统的运行具有不可预测性和不确定性的特点。

概述

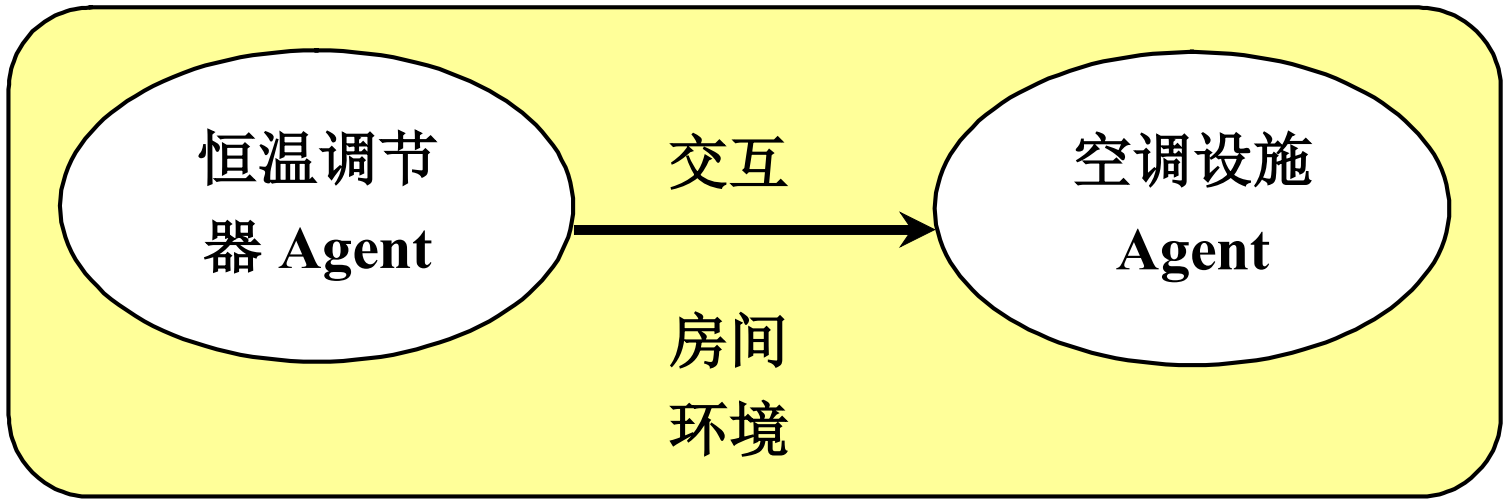


图：多智能体系统的结构

概述

例子1：物理多智能体系统

- 恒温调控系统
  - 恒温调节器Agent
  - 空调设施Agent



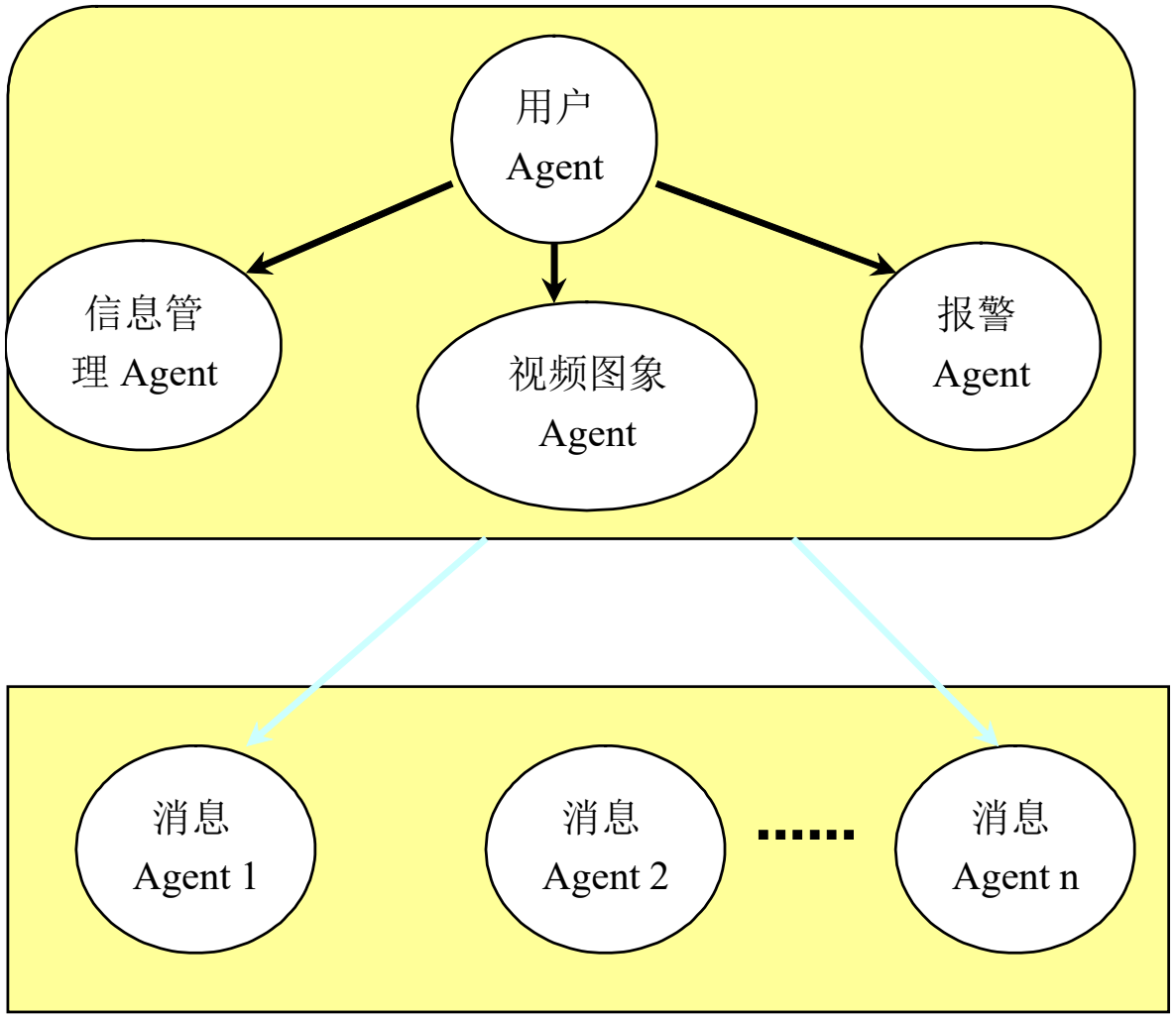
概述

例子2：软件多智能体系统

- 计算机病毒检测和清除软件系统
  - 系统包含的Agent
    - 文件实时防护Agent
    - 病毒分析和清除Agent
    - 病毒数据维护Agent

概述

例子3: 家庭智能网络



目录

- 概述
- Agent及其结构
- 协调与协作
- 基于大语言模型的Agent

Agent及其结构

Agent定义

- 所谓Agent是指驻留在某一环境下，能够自主（Autonomous）、灵活（Flexible）地执行动作以满足设计目标的行为实体。
- 上述定义具有如下两个特点。

(1)定义方式。Agent概念定义是基于Agent的外部可观察行为特征，而不是其内部的结构。

(2)抽象层次。Agent概念更加贴近于人们对现实世界（而不是计算机世界）中行为实体的理解。

Agent及其结构

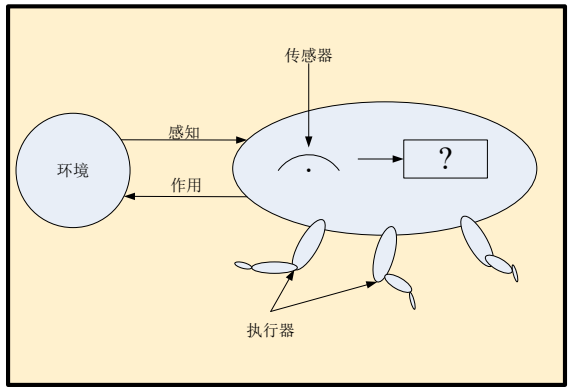
Agent要素及特性

1. Agent的要素

Agent必须利用知识来修改其内部状态(心理状态)，以适应环境变化和协作求解的需要。

人类心理状态的要素可分为以下三种

- 认知(信念、知识、学习等)
- 情感(愿望、兴趣、爱好等)
- 意向(意图、目标和承诺等)



受人类心理启发，传统Agent理论模型研究的主要方向为：研究信念(belief)、愿望(desire)和意图(intention)的关系及其形式化描述，力图建立Agent的BDI(信念、愿望和意图)模型。

Agent及其结构

Agent要素及特性

- 2. Agent的特性
- Agent与分布式人工智能系统一样具有协作性、适应性等特性。此外，Agent还具有自主性、交互性以及持续性等重要性质。

(1) 行为自主性
(2) 作用交互性（也称反应性）
(3) 环境协调性
(4) 面向目标性
(5) 存在社会性
(6) 工作协作性
(7) 运行持续性
(8) 系统适应性
(9) 结构分布性
(10) 功能智能性

Agent及其结构

Agent的结构特点

- 计算机系统为Agent的开发和运行提供软件和硬件环境支持，使各个Agent依据全局状态协调地完成各项任务。具体地说：

（1）在计算机系统中，Agent相当于一个独立的功能模块、独立的计算机应用系统，它含有独立的外部设备、输入输出驱动装备、操作系统、数据结构和相应的输出。

（2）Agent程序的核心部分叫做决策生成器或问题求解器，起到主控作用。

（3）Agent运行是一个或多个进程，并接受总体调度。

（4）各个Agent在多个计算机CPU上并行运行，其运行环境由体系结构支持。

Agent及其结构

Agent的结构分类

（1）**知识型体系结构：**

- 具有表示智能体状态的部件
- 状态表现为智能体所具有的知识
- 智能体的行为决策是一个基于其知识的定理证明过程。

（2）**反应型体系结构：**

- 不具有表示智能体状态的部件，而是定义了一系列的反应式规则
- 智能体根据感知到的环境信息，触发相应的反应式规则来执行，因而其动作决策实际上是一个从情景到动作的直接映射。

（3）**认知型体系结构：**

- 具有表示智能体状态的部件
- 状态表现为智能体所具有的认知状态，如信念、期望、意图等
- 智能体的自主行为决策是一个针对其内部认知状态的实用推理（Practical Reasoning）过程。

目录

- 概述
- Agent及其结构
- 协调与协作
- 基于大语言模型的Agent

协调与协作

多智能体系统的主要目标：

- 以自主的智能Agent为中心，使多Agent的知识、愿望、意图、规划、行动协调，以至达到协作。

协调与协作

概念区分：协调 & 协作

- 协调是指一组智能Agent完成一些集体活动时相互作用的性质。
- 协调是对环境的适应。在这个环境中存在多个Agent并且都在执行某个动作。
- 协调一般是改变Agent的意图，协调的原因是出于其他Agent的意图存在。

- 协作是非对抗的Agent之间保持行为协调的一个特例。多Agent是以人类社会为范例进行研究的。

协调与协作

- 人类交互一般在纯冲突和无冲突之间。
- 在开放、动态的多Agent环境下，具有不同目标的多个Agent必须对其目标、资源的使用进行协调。
  - 如，在出现资源冲突时，需要协调。单个Agent无法独立完成目标，需要协作。

协调与协作

- 在多Agent系统中，协作能：
  - 提高单个Agent以及由多个Agent所形成的系统的整体行为的性能
  - 增强Agent及Agent系统解决问题的能力
  - 使系统具有更好的灵活性
- Agent间的相互依赖关系对Agent的设计和实现具有相当大的制约性
- 取决于不同的交互及协作机制，多Agent系统中的Agent的实现方式将各不相同

协调与协作

Agent协作：

(1) 将其他领域（如博弈论、经典力学理论等）研究多Agent行为的方法和技术用于Agent协作的研究；

(2) 从Agent的目标、意图、规划等心智态度出发来研究多Agent间的协作，如FA/C模型、联合意图框架、共享规划等。

所运用的各种理论只适用于特定的协作环境下，而一旦环境发生变化，如Agent的个数及Agent间的交互关系与该理论所适用的情形不一致时，基于该理论的协作机制就失去了其存在的优势。

较偏重于问题的规划与求解，并且它们所假定的协作过程差异显著。
（1）先找协作伙伴再规划求解，
（2）先对问题进行规划然后由Agent按照该规划采取协作性的行动，
（3）在Agent自主行动的过程中，进行部分全局规划来调整自己的行为以达到协作的目标。

协调与协作

- 在多Agent系统中，Agent是自主的，多个Agent的知识、愿望、意图和行为等往往各不相同，对多个Agent的共同工作进行协调，是多Agent系统的问题求解能力和效率得以保障的必要条件。

  - 包括组织理论、政治学、社会学、社会心理学、人类学、法律学、以及经济学等在内的多个学科领域都对协调进行了研究，许多研究成果已经应用到多Agent系统中。
  - 多Agent系统中的协调是指多个Agent为了以一致、和谐的方式工作而进行交互的过程。

协调与协作

进行协调是希望避免Agent之间的死锁和活锁。

死锁指多个Agent无法进行各自的下一步动作

活锁是指多个Agent不断工作却无任何进展的状态

- 多Agent之间的协调方法：
  - 组织结构化
  - 合同
  - 多Agent规划
  - 协商

协调与协作

多Agent之间的协作可分为（社会心理学）：

- 协作型：同时将自己的利益放在第二位；
- 自私型：同时将协作放在第二位；
- 完全自私型：不考虑任何协作；
- 完全协作型：不考虑自身利益；
- 协作与自私相混合型。

协调与协作

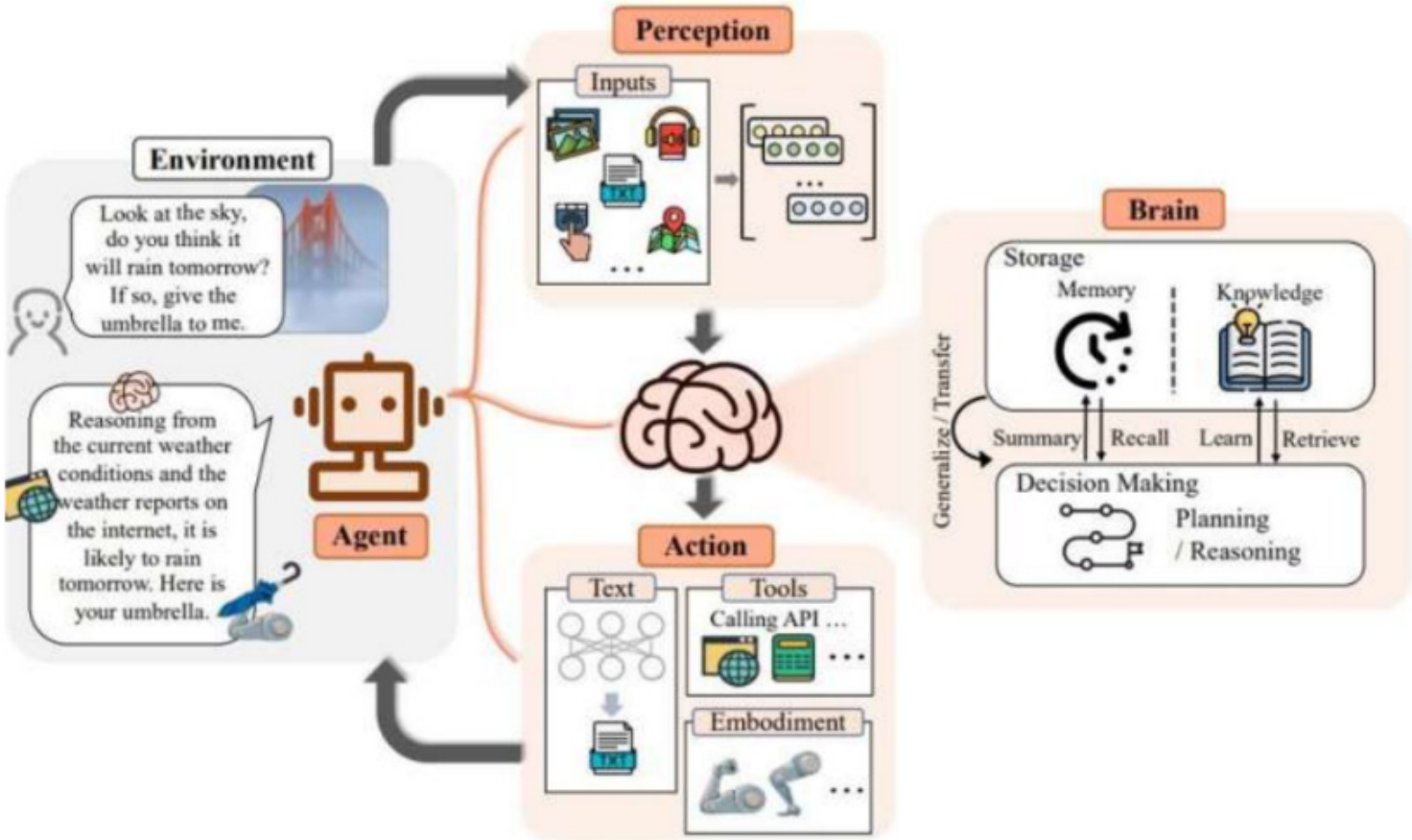
- 现实世界是开放的、动态的，协调与协作也应是开放的、动态的。
- Agent在开放的、动态的环境中不一定具备很强的推理能力，而可以通过不断的交互，逐步协调与环境以及各自之间的关系，使整个系统体现一种进化能力。
- 在BDI模型中则强调交互作用中Agent信念、愿望和意图的理性平衡。

目录

- 概述
- Agent及其结构
- 协调与协作
- 基于大语言模型的Agent

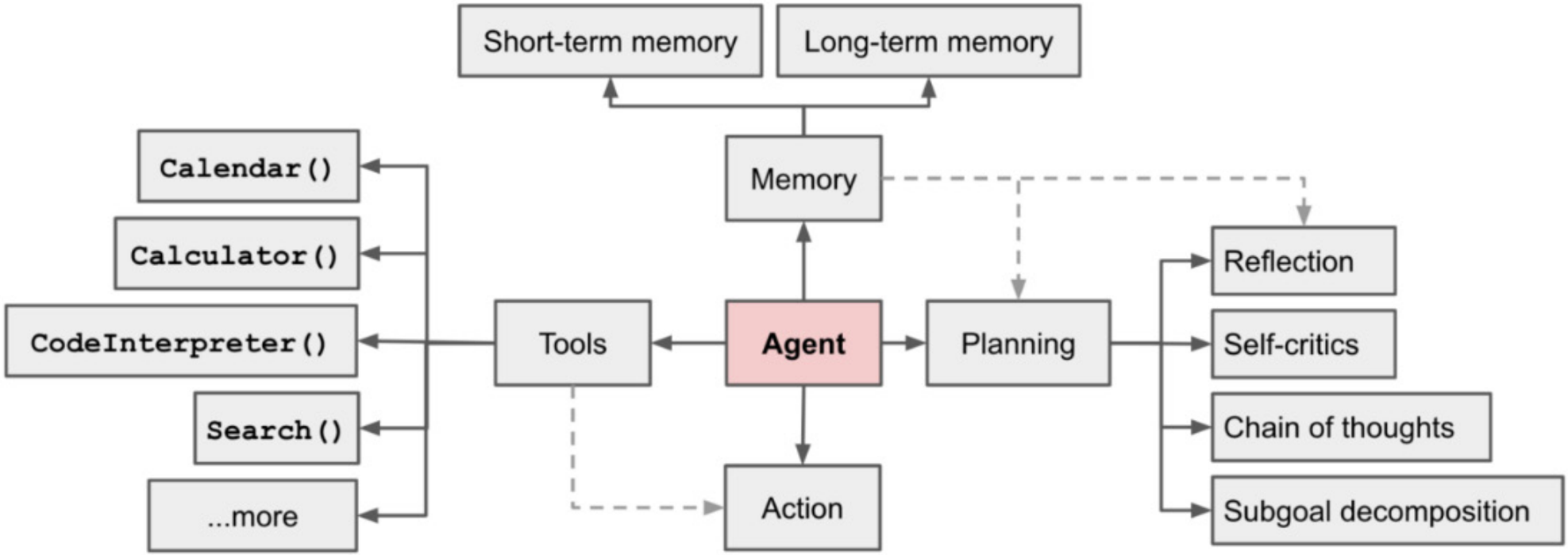
基于大语言模型的Agent

LLM-based Agent的概念框架包含三个组成部分：
控制端（Brain）、感知端（Perception）和行动端（Action）



基于大语言模型的Agent

Agent内部核心框架



基于大语言模型的Agent

Planning（规划）

基于大语言模型的Agent

任务分解 Task Decomposition

- Chain of thought https://lilianweng.github.io/posts/2023-03-15-prompt-engineering/#chain-of-thought-cot
- Tree of thought https://arxiv.org/abs/2305.10601
- LLM+P https://arxiv.org/abs/2304.11477
- Calibrating trust of multi-hop question answering systems with decompositional probes
- Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions

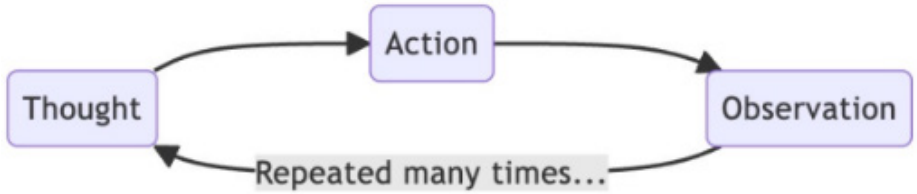
基于大语言模型的Agent

自我反思 Self-Reflection

- ReAct https://arxiv.org/abs/2210.03629

Thought: ...
Action: ...
Observation: ...
... (Repeat)

通过改进过去的行动决策和纠正错误来实现迭代改进



HotpotQA

(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple … originally designed to control the Front Row media center program …

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports',' Front Row (software) ', …]

Thought 3: Front Row is not found. I need to search Front Row (software) .
Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software …

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.
Act 4: Finish[keyboard function keys]

AlfWorld Env

(2b) ReAct (Reason + Act)

Act 1: Think[First I need to find a pepper shaker...more likely to appear in cabinets (1-6), countertops (1-3), ...]
Act 2: Go to cabinet 1
Obs 2: On the cabinet 1, you see a vase 2.
(Here the agent go to cabinet 1, then cabinet 2, 3, then countertop 1 and 2)
Act 6: Go to countertop 3
Obs 6: On the countertop 3, you see a apple 1, a bread 1, ... a pepper shaker 1, and a vase 3.

Act 7: Take pepper shaker 1 from countertop 3
Obs 7: You pick up the pepper shaker 1 from the countertop 3.

Act 8: Think[Now I find a pepper shaker 1. Next, I need to put it in/on drawer 1.]
Act 9: Go to drawer 1
Obs 9: Drawer 1 is closed.

Act 10: Open drawer 1
Obs 10: You open Drawer 1 …

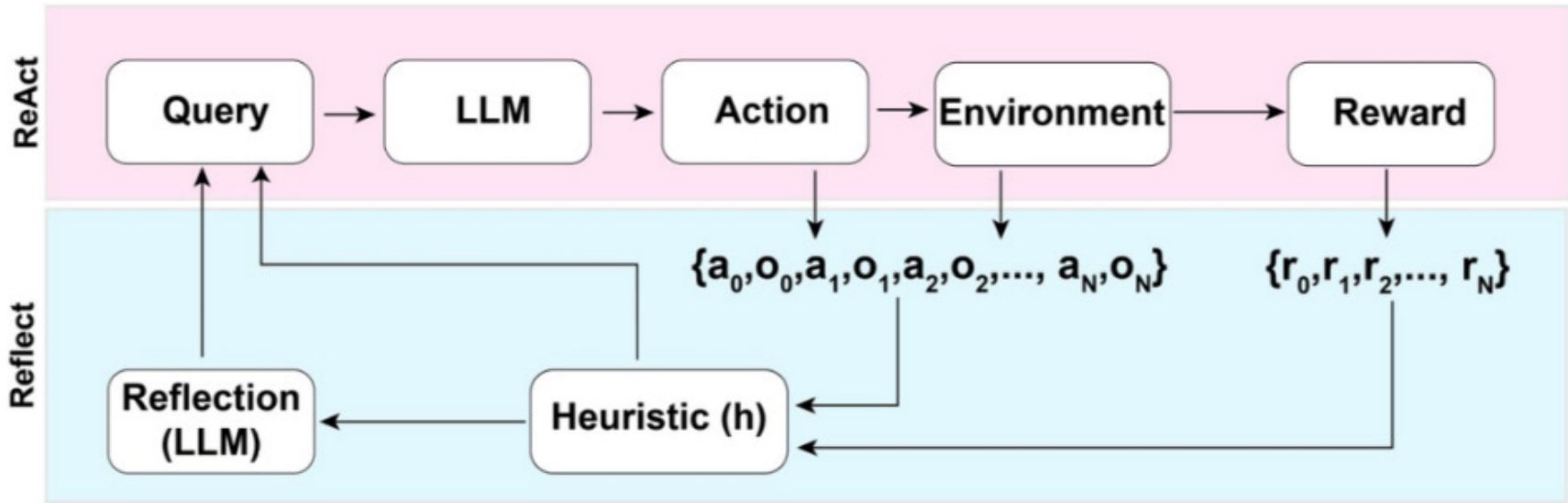
Act 11: Put pepper shaker 1 in/on drawer 1
Obs 11: You put pepper shaker 1 in/on the drawer 1.

基于大语言模型的Agent

自我反思 Self-Reflection

- ReAct https://arxiv.org/abs/2210.03629

动态记忆和自我反思能力



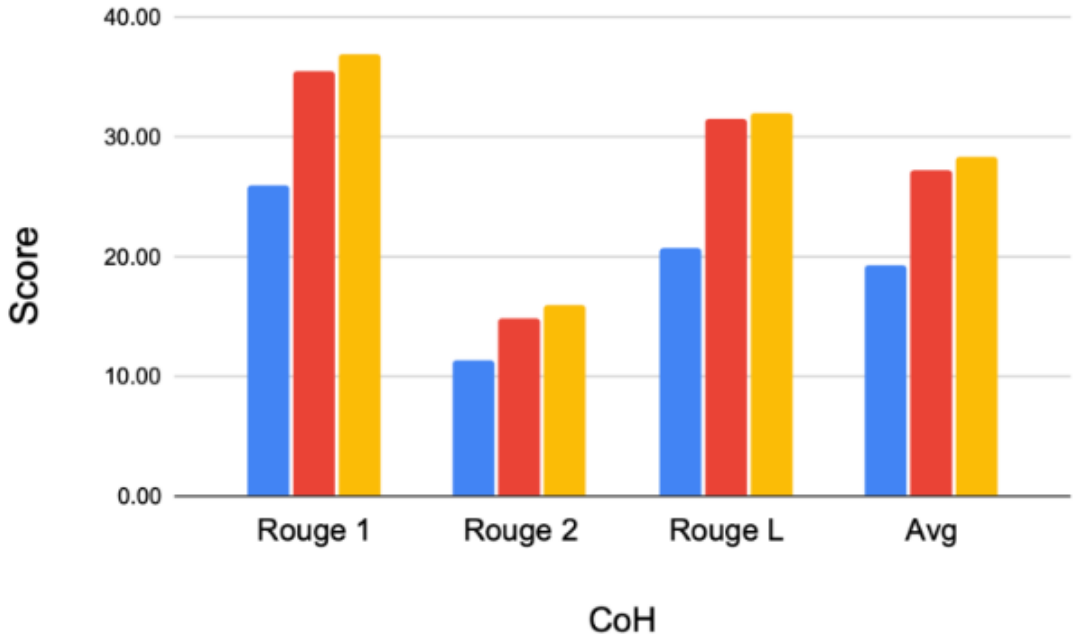
基于大语言模型的Agent

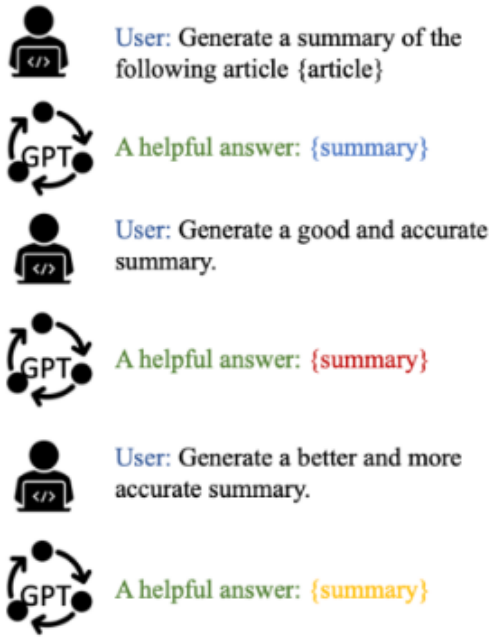
自我反思 Self-Reflection

- 回溯链Chain of Hindsight

https://arxiv.org/abs/2302.02676

不断给予模型过去的表现反馈，包括评分和建议；向模型展示一系列不断进步的答案，然后训练模型学习这种进步的趋势，从而得到更好的答案。

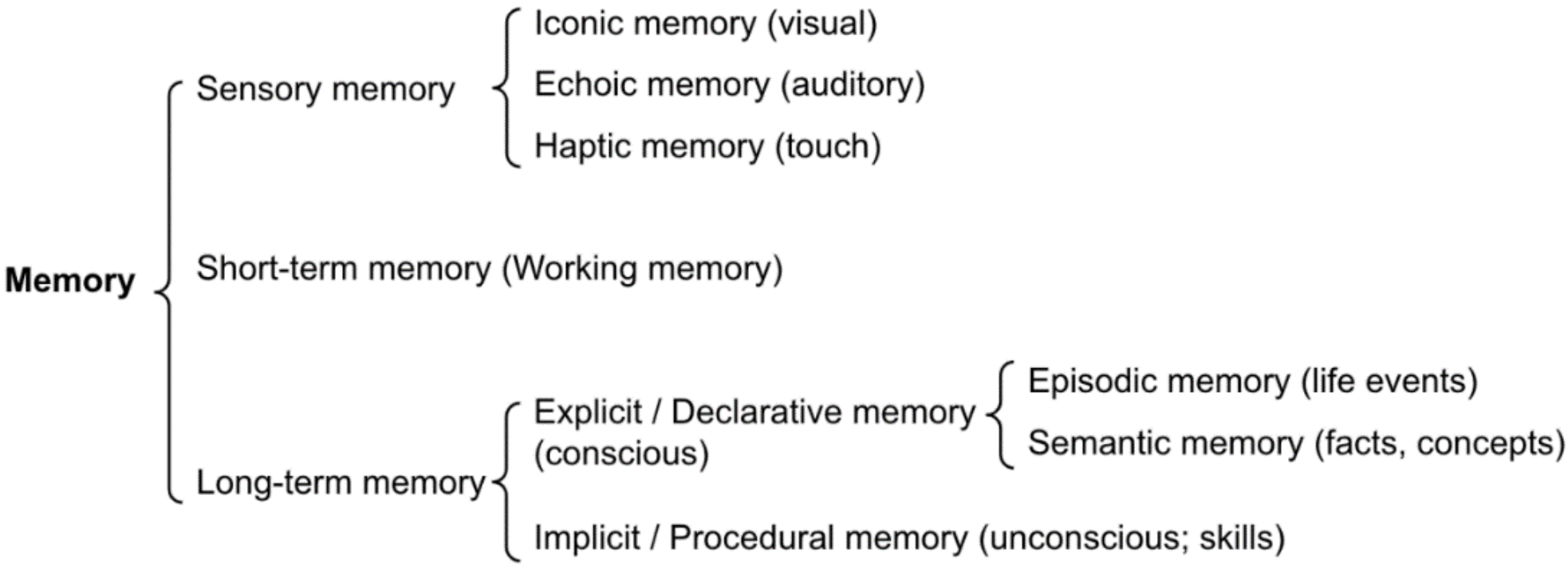




基于大语言模型的Agent

Memory（记忆）

基于大语言模型的Agent



基于大语言模型的Agent

Tool Use（工具使用）

基于大语言模型的Agent

大模型：语言理解力强，缺点：模型出现幻觉，需要精心设计prompt，需要依赖外界工具

Agent：大模型+各式各样的工具辅助

外界专业工具

prompt：可以使用外部工具

```
...
你可以使用如下工具来完成任务：

1. 计算器，用来执行各种数学计算获取精确结果，输入表达式，例如 1 + 1，得到结果
...

问题：123 乘以 456 的结果是多少？
...
```

模型生成的内容如下：

调用外部计算器

```
思考：我需要使用计算器来计算 123 乘以 456 的结果
动作：调用计算器
动作输入：123 * 456
观测结果：
```

扬长：大模型的语言理解能力可以充分理解指令和任务

避短：专业的计算交给专业的工具来做

基于大语言模型的Agent

Agent项目

- Auto-GPT https://github.com/Significant-Gravitas/Auto-GPT
- LangChain https://github.com/langchain-ai/langchain
- AutoGen https://github.com/microsoft/autogen
- GPT Engineer https://github.com/AntonOsika/gpt-engineer
- BabyAGI https://github.com/yoheinakajima/babyagi
- AI Town https://github.com/a16z-infra/ai-town
- GPTeam https://github.com/101dotxyz/GPTeam
- ChatArena https://github.com/chatarena/chatarena
- AgentVerse https://github.com/OpenBMB/AgentVerse

基于大语言模型的Agent

小结

如果说大模型是电池：核心能力
那么agent就是一辆电动车：最终交付，最终产品

Agent的缺点：

1.依赖大模型的核心能力，大模型本身够强才行
2.链路过长，某一环节出错，前功尽弃
3.多次调用模型，效率不高
4.迁移能力弱，换模型需要重新写提示词
5.能力强弱，取决于写提示词的水平

展望

大模型能力是有上限的
端到端虽然理想，但是复杂问题很难解决
Agent这种拆解形式，今后会更加流行？