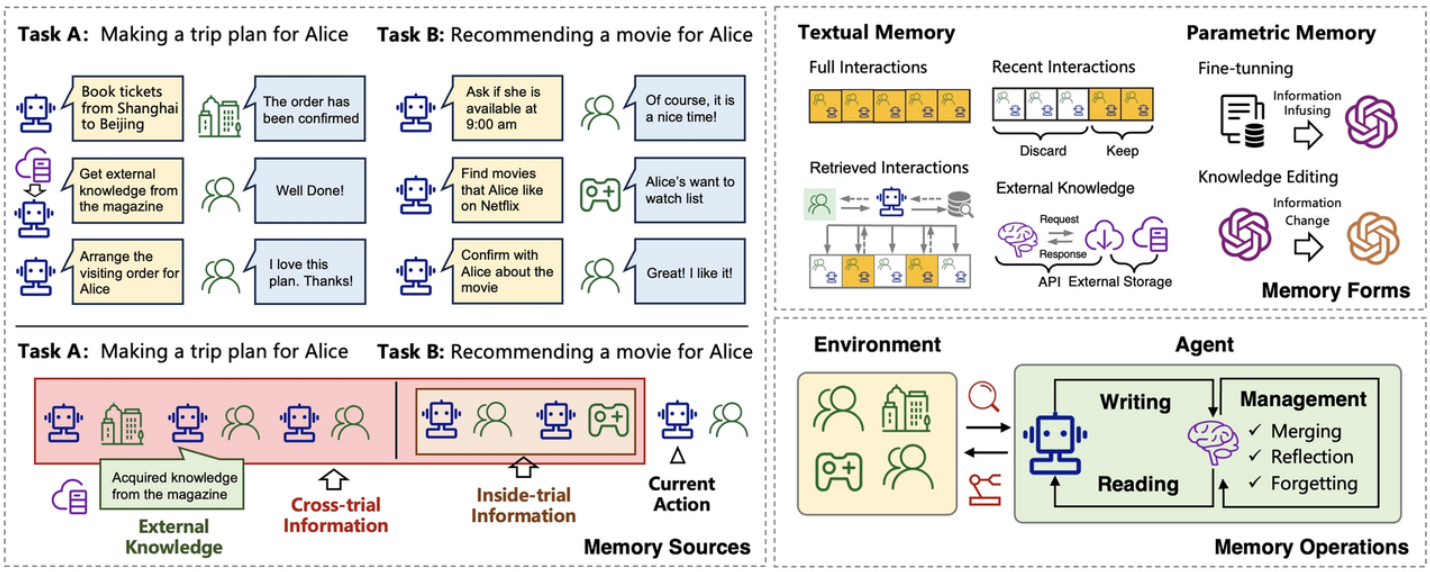
Memory-based agent2

如何实现Memory-based agent



这张图是对Agent中memory的来源、形式、操作方法的概述。

记忆来源

如上一篇笔记介绍的，memory来源于三种情况：**同一个trial，跨trail跨任务，外部知识**。

其中，**同一个trial**和**跨trail跨任务**都来自agent与环境交互，**外部知识**来自任务以外的环境。

- **同一个trial**：同一个trial中的历史步骤是最相关最有信息量的信息，与agent的任务高度相关。

> 这部分信息不仅包括代理环境交互，还包含交互上下文，例如时间和位置信息。

- **跨trial跨任务**：在环境中多次试验所积累的信息也是记忆的重要组成部分，通常包括成功和失败的操作及其见解，例如失败的原因、成功的常见行动模式等，基于过去的经验，agent可以根据整个过程的整体反馈调整其行为。代表工作是Planning篇介绍的Reflexion。同一个trial的可以视为短期记忆，跨trial跨任务的则是长期记忆。
- **外部知识**：根据任务需求实时动态访问各种工具的 API 来获取，例如搜索维基百科，扩展agent的知识边界。代表工作是Planning篇介绍的React。

记忆形式

记忆一般有两种形式：**文本形式**和**参数化形式**。

- **文本形式**是目前表示记忆内容的主流方法，具有可解释性更好、更容易实现、读写效率更快的特点。过去的研究主要分为4个方向：

- （1）完整的agent-环境交互，计算成本和推理时间都会大幅增加，并且推理容易不鲁棒，因为文本在长上下文中的位置会极大地影响它们的利用率，长上下文提示中的记忆不能得到平等和稳定的处理。
- （2）最近的agent-环境交互，会遗忘遥远的记忆，对于长期任务可能导致关键信息缺失，并且如何确定缓存窗口也比较困难。
- （3）检索到的agent-环境交互，根据记忆内容的相关性进行检索，在记忆读取时，会为每个记忆条目计算匹配分数，并选取 top-K 条目。在记忆写入时生成embedding作为记忆的索引，以辅助信息检索。检索需要额外的计算成本。
- （4）外部知识，通过调用工具获取外部知识（如维基百科），可能会涉及检索、重排等多个计算过程，如何将外部知识和内部决策过程融合值得研究。

- **参数化形式**的记忆存储在模型的参数中，不受 LLM 上下文长度限制的限制。现有方法主要分为两类：微调方法和记忆编辑方法。
  - （1）微调方法：通过有监督微调将外部知识整合到代理的记忆中，能显著提升模型在特定领域的能力。只能应用在离线场景。面临着微调通用的问题，灾难性遗忘，过拟合，需要大量数据，计算成本高等。
  - （2）记忆编辑方法：直接修改Transformer模型的权重,以实现对模型记忆的精确编辑。记忆编辑只针对需要修改的事实，更适合小规模的记忆调整，适合在线场景。

对比**文本形式记忆**和**参数化形式记忆**：

- 有效性。文本记忆存储有关agent与环境交互的原始信息，这些信息更全面、更详细。然而，它受到 prompt中token上限的限制，这使得agent难以存储大量信息。相比之下，参数记忆不受提示长度的限制，但在将文本转换为参数时可能会遭受信息丢失，复杂的记忆训练会带来额外的挑战。
- 效率。对于文本记忆，每次 LLM 推理都需要将记忆整合到上下文提示中，这会导致更高的成本和更长的处理时间。相比之下，对于参数记忆，信息可以集成到 LLM 的参数中，从而消除了这些上下文的额外成本。然而，将参数记忆写入模型的需要额外的成本，但文本记忆更容易写作，尤其是对于少量数据。简而言之，文本记忆在写作方面更有效率，而参数记忆在阅读方面更有效率。
- 可解释性。文本记忆通常比参数记忆更易于解释，因为自然语言是人类理解的最自然、最直接的策略，而参数记忆通常以潜在空间表示。然而，这种可解释性是以信息密度为代价的。这是因为文本记忆中的单词序列以离散空间表示，而离散空间不像参数记忆中的连续空间那样密集。
- 总之**文本记忆**更适合对话和特定上下文的场景，并且上下文内容不太多。**参数化记忆**适合有大量知识需要写入模型记忆的场景。

记忆操作

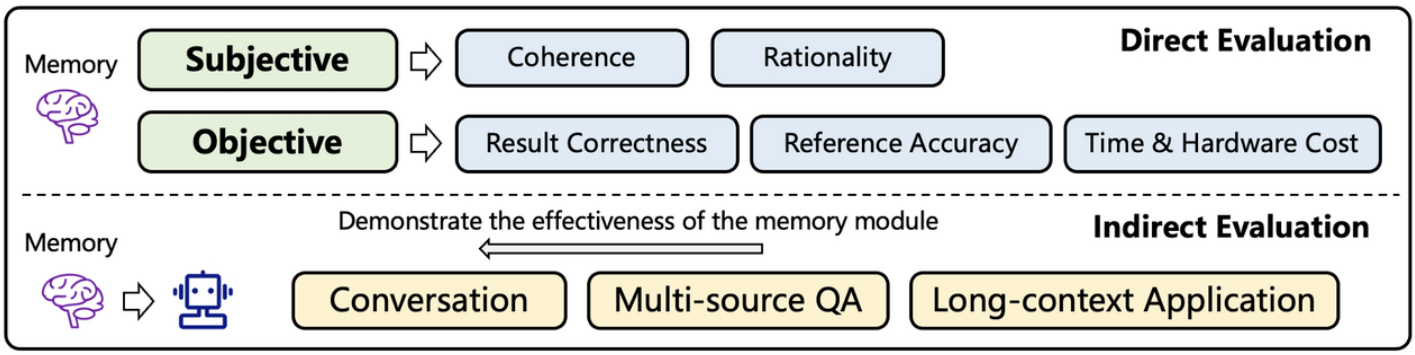
如上文所述，对记忆的操作主要有**写记忆，记忆管理，读记忆**

- **写记忆**：识别哪些信息应该写入记忆至关重要，并且应该对原始信息中的噪声进行处理。此外，环境可能会提供各种形式的反馈，如何从这些反馈中提取有效信息（可能涉及多模态）也至关重要。
- **记忆管理**：通过不断的反思来管理，以生成更高级的记忆，合并多余的记忆条目，并忘记不重要的早期记忆。
- **读记忆**：对于文本形式的记忆，可以采用文本相似来筛选记忆。而对于参数化的记忆，模型直接用更新后的模型进行推理，这可以视为隐式的阅读过程。

如何评估Memory-based agent

本文将评估方法分为两类：（1）直接评估，独立测量记忆模块的能力。（2）间接评估，通过端到端代理任务评估记忆模块。

- **直接评估**：又可以分为人的主观评价和客观指标评价两种
  - 主观评价：主要从回答的连贯性，即检索到的记忆应该与当前情景匹配，以及合理性，即检索到的以及应该包含正确的信息。
  - 客观评价：常见指标包括预测标签准确率，检索到的记忆的F1值，计算和推理时长，GPU占用量
- **间接评估**：如果代理能完成高度依赖记忆的任务，那就证明记忆模块是有效的。代表性任务有：
  - 对话任务：记忆保留与用户对话的上下文，agent的回复应该与上下文一致，不能出现自相矛盾的情况。还可以通过用户的参与度来衡量agent回复的质量和吸引力。
  - 多源问答：综合评估之前提到的多个来源的记忆信息，即同一个trial、跨trail和外部知识，测试agent对不同内容和来源的记忆的整合能力。同时检测多信息源导致的记忆矛盾和知识更新问题。
  - 长文本应用：采用文本形式的记忆会一般都会导致比较长的上下文，需要评估agent能否充分理解长文本，并从中检索出对给定问题有用的内容。



参考：**A Survey on the Memory Mechanism of Large Language Model based Agents**