**KwaiAgents: Generalized Information-seeking Agent System with Large Language Models**

KwaiAgents: 基于大型语言模型的通用信息搜索代理系统

**太长不看版：**

在这篇论文中，介绍了KwaiAgents，它由三个核心组件组成：系统、llm和基准测试。KwaiAgents是一个基于大型语言模型（LLMs）的通用信息搜索代理系统。这个系统使用LLMs作为其认知核心，旨在理解和响应用户的需求，并从各种来源中检索和整理信息。KAgentSys代理系统采用了一种规划总结流程，利用大型语言模型（LLMs）作为核心认知单元、一个记忆库和一个混合时间感知搜索浏览工具包，以有效地完成用户查询。为了研究开放源码、较小规模的LLMs在展示代理能力（如规划、反思和工具利用）方面的潜力，作者提出了元代理调整（MAT）策略，并制作了一套名为KAgentLMs的LLMs。此外，作者还建立了一个全面的基准测试KAgentBench，以评估上述LLMs的能力，并收集了约200个查询，以进一步评估系统层面的性能。广泛实验表明，作者的系统优于其他开源代理系统。经过MAT处理后，即使是6B-13B范围内的LLMs也表现出了与GPT-4相当的结果。

**摘要**

在好奇心的驱使下，人类不断地寻求探索和理解他们周围的世界，这导致了各种工具的发明来满足这种好奇心。尽管人类没有能力处理和记忆大脑中的大量信息，但他们擅长于批判性思维、计划、反思和利用可用的工具来与世界互动和解释世界，使他们能够有效地找到答案。大型语言模型（*llm*）的最新进展表明，机器也可能具有上述的类人能力，这使它们即使在参数计数受限的情况下也能表现出强大的能力。本文介绍了一种基于*llm*的广义信息搜索代理系统*kwaiet*代理。在*KwaiAgents*中，作者提出了一个使用*llm*作为其认知核心的代理系统，它能够理解用户的查询、行为准则和引用外部文档。代理还可以从其内部内存中更新和检索信息，并使用可感知时间的搜索浏览工具包来计划和执行操作，并最终提供综合的响应。作者进一步研究了系统在不如*GPT-4*先进的*llm*驱动下的性能，并引入了*Meta-Agent Tuning*优化（*MAT*）框架，该框架旨在确保开源的7B或13B模型在许多代理系统中表现良好。作者利用基准测试和人工评估来系统地验证这些能力。大量的实验表明，作者的代理系统比其他自治代理的优越性，并体现了作者的微调后的*llm*的增强通用代理能力。

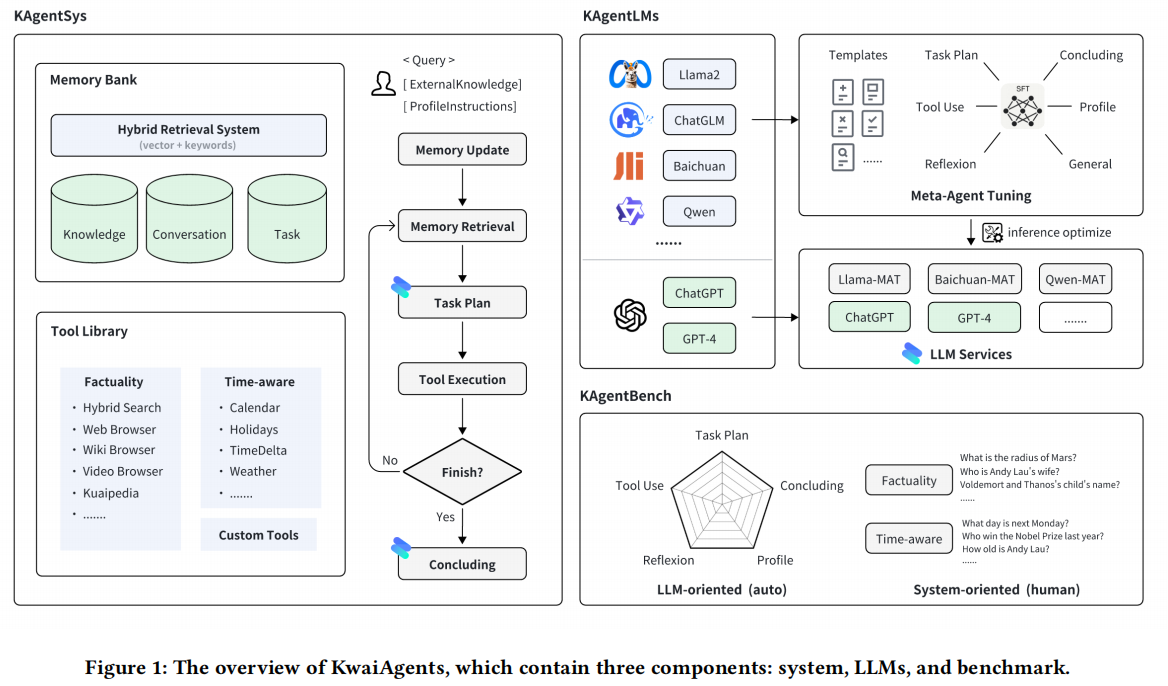


图1：KwaiAgents的概述，它包含三个组件：系统、llm和基准测试。

1. **引言**
2. **系统**

在本节中，将描述KAgentSys的功能及其对用户需求的响应 。首先解释llm、内存库和工具库的作用。随后，这些组件将被集成到KAgentSys的主代理循环中。

* 1. **LLMs**

与人脑类似，llm应该首先了解用户的需求。这包括考虑与从数据库中检索到的当前对话轮次相关的外部知识，以及引用以前的会话消息和过去已完成的任务。随后，LLM应制定有充分根据的计划、适当的工具命令，或在完成规划过程后得出结论。为了适应不同llm的不同需求，我们引入了一个简单的调用API，它接受一个提示符作为输入并返回一个响应。

* 1. **存储库**

内存模块在整个会话过程中缓存用户的上下文信息。这些信息可分为三个不同的组成部分：

**知识记忆：**此模块捕获并检索用户希望集成到会话上下文中的外部资源。例如，用户希望详细讨论的个人数据或文档。

**对话记忆：**此组件记录对话中每个回合的“查询-响应”。

**任务记忆：**这段记忆记录了KAgentSys的决策过程。对于每个会话，一旦用户输入文本，KAgentSys就会通过规划任务、选择适当的工具和执行命令从这些工具中获取结果。最后，这些与任务相关的信息被系统地存储在记忆中。

对于每种内存类型文本被划分为具有固定最大长度的段。然后将每个段转换为基于向量的搜索的向量表示，或使用反向列表的关键字搜索进行索引。这在一个混合检索系统中达到了顶点，给定一个简洁的查询，使用不同的检索机制从不同的内存类型中提取相关的文本片段。然后将这些段聚合并格式化为结构化的、上下文丰富的文本，以便在后续提示中使用。

* 1. **工具库**

该工具库提供了两套不同的预定义工具：一套用于事实性，另一套用于时间感知。

系统的第一套工具通过引导LLM生成查询来利用搜索引擎进行事实检索。然后，LLM确定是从检索结果中访问特定的URL，还是根据检索到的信息响应查询。如果LLM选择访问一个链接，它将继续扫描该网页，提取相关段落或生成针对该查询的简洁摘要。然而，它将这种方法扩展到“混合搜索”，将传统的网络搜索与实体搜索结合起来。混合\_searcs函数接受单个参数、查询，并同时执行两个任务： (1) 它使用搜索引擎API来获取相关的网页，每个网页都附有一个标题、URL和一个简短的摘要。(2) 它应用主体实体链接来识别主体实体，并收集简明的维基百科描述、方面树和快客百科中最相关的视频。每个组件（实体、方面和视频）都包含其对应的URL，用于详细的探索。当代理选择浏览时，它会选择以下操作之一：(1) browse\_website深入研究一个网页，并对其进行总结或导出与输入查询相关的答案。(2) browse\_wiki浏览维基百科页面，并进行总结或提取与查询相关的答案。(3) browse\_aspect来检查一个实体的方面和调查相关的视频。(4) browse\_video来解释各个帧的OCR和视频的ASR，生成一个基于文本的摘要。

系统的第二套工具强调时间意识。我们观察到，仅仅将时间戳合并到提示中，对于具有时间感知能力的问题回答仍然不是最优的，从而导致过时信息的传播。因此，我们引入了几个可以感知时间的工具，包括： (1) calendar，它处理一个日期范围，以提供来自公历和月历的细节。(2) holidays，给定一个日期范围，会返回显著的节日。(3) time\_delta，它计算两个时间戳之间的持续时间（以天/小时/分/秒为单位）。(4) weather，这需要一个位置和日期范围来提供天气细节，如温度、降水和湿度。此外，用户定义的自定义工具也可以无缝地集成到KAgentSys中，只要它们有详细的规范。

* 1. **代理循环**

在初始化KwaiAgents的循环之前，用户会提交一个查询（必需的），以及一些可选的输入，如外部知识源（链接或文件），以及指导代理行为的特定指令或配置文件。然后，该循环将继续执行以下步骤：

**内存更新：**系统更新用户过去的交互记录，包括以前的对话和已执行的任务。如果用户提供了新的链接或文件，那么外部知识库也会相应地进行更新。

**内存检索：**基于用户当前的查询和最近的交互，可以检索来自外部知识来源、过去的对话和已完成的任务的相关片段。这些段被合并到一个固定最大长度的文本框中以供后续参考。

**任务计划：**这个阶段利用收集到的信息来为llm构建一个全面的提示。提示符包括代理的配置文件、用户指定的行为准则、工具规范、统一内存文本框、当前时间戳以及JSON中的预定义输出格式。因此，LLM将被提示生成一个task\_name，表示随后阶段的计划，以及一个详细说明要调用哪些工具和各自的参数的命令。

**工具执行：**如果“任务计划”阶段生成一个名为task\_fulilte的工具，则循环终止，提示代理制定结论。否则，将执行指定的命令。每个命令都会以自定义格式生成一个观察结果，然后将其集成到任务内存中。

**结论：**类似于“任务计划”阶段，代理通过考虑检索到的内存来对用户的查询制定响应。代理的配置文件、用户提供的指令和当前的时间戳也被合并到最终文件中。

1. **Meata-Agent Tuning**

这段主要讨论了人类和机器在信息处理和知识获取方面的差异和能力。尽管人类的大脑无法处理和记忆大量的信息，但他们擅长批判性思考、规划和反思，并利用可用的工具与世界互动和解释世界，从而能够高效地找到答案。最近的大型语言模型（LLMs）的进展表明，机器也可能具备上述的人类能力，即使参数有限，它们也能表现出强大的能力。

* 1. **模版制作（*templates crafting*）**

主要涉及对多个开源代理系统的系统提示进行审查，并从中提取出六个共同的组件。这些组件包括：(1) 简介：详细说明*LLM*（大语言模型）的角色，例如“你是一个有用的*AI*计划员”。(2) 指令：包含代理的约束和行动顺序，例如“迭代不超过五次”。(3) 工具：概述要使用的工具的格式，例如带有函数名、描述和参数详情的*JSON*模式。(4) 记忆：指示外部知识、过去任务和对话历史的集成，例如将对话历史放在提示的开头。(5) 目标：指定用户查询或要求的整合。(6) 格式：说明代理应如何构建响应和管理迭代，例如返回包含任务名、工具名和参数的*JSON*对象。

此外，该文还描述了如何利用这些见解设计一个meta-agent，使用*GPT-4*来制作模板提示。然而，*GPT-4*生成的模板是同质的，缺乏多样性。为了解决这个问题，引入了特定主题的种子查询来指导*GPT-4*产生更有针对性和多样化的模板。因此，查询收集在此上下文中至关重要。从各种来源收集高质量的查询，包括*ShareGPT 3*、先前的研究和基于工具的*GPT-4*生成查询，遵循既定方法。在编译查询后，将它们分成重叠的群集，如名人信息（如图2所示）。最后，meta-agent帮助生成一系列系统提示模板供后续使用。

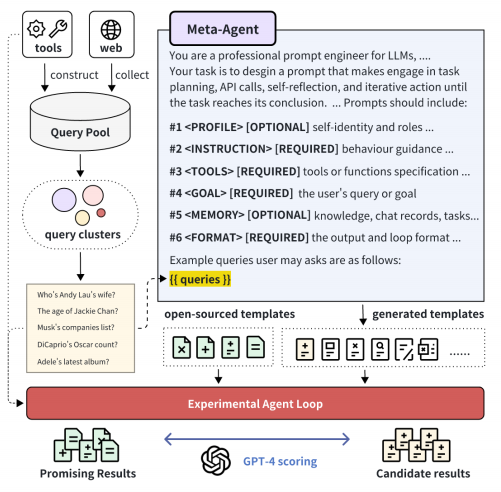


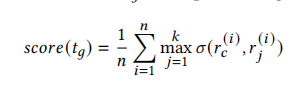
图2：元代理如何构建代理系统模板和验证的过程。

* 1. **模版验证**

这段话主要讨论了模板生成的主要问题，即输出的质量。尽管*GPT-4*具有先进的功能，但生成的模板往往存在重大缺陷，无法在模拟环境中运行或与预期的人类结果不符。为了解决这个问题，一个验证框架对于评估和确定由*GPT-4*生成的代理系统模板的可行性至关重要。

分析表明，开源代理系统模板通常比*GPT-4*生成的模板具有更高的成功率。通过捆绑这些经过验证的模板，我们可以加强验证过程。这个过程包括一个比较分析，其中将一组查询Q应用于*GPT-*生成的模板*tg*和开源模板*T0*。这些模板的性能随后在由*GPT-4*驱动的实验代理循环中进行评估，产生一系列有前途的结果*R*。同时，循环为tg生成一个候选结果*rc(i)*。我们利用*GPT-4*的分析能力来评估*rc(i)*与有前途的结果之间的接近程度。

模板的最终得分使用以下公式计算：



最后，我们设定一个阈值e，批准满足或超过该基准的模板*tg*，即*score(tg)≥e*。

* 1. **LLMs 训练(*LLMs training*）**

首先，经过初始阶段，我们已经积累了大量高质量的生成代理系统模板、开源模板和在*KAgentSys*中使用的模板。接下来，我们对每个模板进行了一系列的查询，使用了一个实验代理循环，该循环提示*GPT-4*生成相应的提示-响应对。这一过程产生了代理数据集*Dagent*，该数据集与多样化的指令数据集*Dgeneral*合并，用于训练基础的大型语言模型。

此外，我们还尝试了适当的混合率η，该混合率在这些两个数据集之间应用，以进行训练阶段。混合率η决定了在训练过程中使用哪个数据集的比例。通过调整这个混合率，可以平衡模型的泛化能力和特定领域的知识。

具体的数据集构建和分布细节可以在附录B中找到。这段话主要关注数据集的生成和大型语言模型的训练过程，以及如何将不同来源的数据集合并在一起进行训练。

1. **实验**

具体涉及到了特别策划的指令调优数据集和*KAagentBench*的介绍。文章还展示了不同大型语言模型(*LLMs*)在我们的基准测试上的表现，并从不同角度说明了模型在代理能力方面的差异。此外，文章还收集了一些真实或时间感知的查询，以展示不同模型在不同系统上的表现。

* 1. **数据集**

首先，提到了如何编制指令调优数据集。数据集通过汇总和生成来自先前研究的工具和查询来创建，并进行了扩充，添加了聚焦于事实内容的额外查询。在模板制作阶段后，从每组生成中抽取了一些查询，以产生大量候选模板。然后，通过实验代理循环对这些模板进行审查，只有达到阈值分数的模板才会被选中并使用。数据集包括120,917个查询和18,005个模板，这些模板来自开源*KAgentSys*和生成的内容。此外，为了确保不过分强调*LLMs*的专门能力，还从不同来源收集了43,099个来自不同领域的查询。

其次，详细介绍了*KAgentBench*的构建过程。这个基准测试通过严格选择查询、工具、模板和记忆元素来创建，其详细分解在图2中展示。基准测试中63.04%的查询是中文，其余为英文。每个查询都与五个系统提示模板配对，但“*Profile*”类型除外，它增加了实例数量。每个实例都有四到五个由不同模板生成的真值，前提是这些真实性与提示模板无关。基准测试共有614个独特的工具，平均每个查询有9.26个定制的工具。记忆分为四类：无、对话轮次、任务历史或外部知识。其中，无记忆占20.64%，对话轮次占48.12%，任务历史占16.23%，外部知识占15.01%。此外，对于涉及到的记忆、对话轮次、任务历史和外部知识数据，都包含了与查询有关、无关和冲突的信息类型。其中，“*Reflection*”需要解决前任务或外部知识中的错误信息或差异，要求*LLMs*重新评估当前信息以进行更策略性的规划，如制定替代的搜索查询。“*Profile*”维度涉及在特定角色中创建教学对话，*GPT-4*在不同的设置下生成变化。最后，人工标注对每个标签进行了细化，以确保高质量的基准测试。

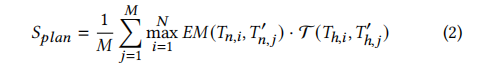
此外，还收集了200个中文的事实和时间感知查询，作为代理系统的初始输入。还可以选择使用人类注释器来评估系统最终响应的准确性和质量。

* 1. **基准评价(*benchmark evaluation*)**

**第一**，描述了在规划阶段，模型需要生成三个关键输出，并介绍了用于评估不同代理能力的特定数据集和指标。下面我将详细解释这些内容：

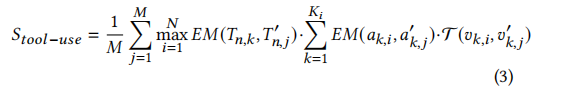
首先，模型在规划阶段的任务是产生三个关键输出：*thought Th、tool-name Tn、tool-args Ta。*

**规划能力评估：**评估涉及遵循“知识与实践的统一”原则。这需要同时考虑*thought*和*tool-name*。评估公式为：



其中，T∗,i是第i个真值结果，T∗′,j是第j个提示模板预测的结果。M表示模板的数量，N表示真值结果的数量。T是用于文本生成评估的任何指标，例如ROUGE分数。

**工具使用熟练度评估：**通过工具选择（*tool-name*）和工具应用（*tool-args*）来衡量。评估公式为：



**第二**，描述了不同评估阶段的计算方法和指标，涉及规划、工具使用、结论生成和角色指导对话等方面。

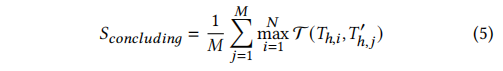
**规划阶段：**

评估规划能力时，使用一个特定的数据集。评估考虑了*thought*和*tool-name*，并结合了规划得分方法和工具使用熟练度评估。如果生成的*tool-names*和*arguments*与之前的任务重复，将应用一个惩罚因子*p*。规划得分的计算公式为：



**结论生成阶段：**

结论是一个单词序列，因此评估相对简单。结论得分的计算公式为：



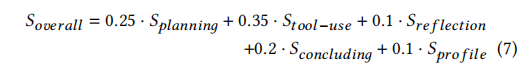
**角色指导对话：**

使用一个特定的数据集进行基准测试，主要涉及基于角色的指导性对话。对于这种对话的响应，也是一个单词序列，但只有一个模板和一个地面真实值。得分的计算公式为：



**综合评估：**

综合得分是对上述五个指标的加权和。权重的选择需要仔细考虑。综合得分的计算公式为：



**第三**，对不同LLM的性能进行了评估，并与前两段内容有所关联。以下是具体的分析：

**整体性能比较：**

文中提到，在所有实验的开源模型中，GPT-3.5的性能最佳。

**特定模型性能变化：**

经过*meta-agent*调整后，*Qwen-7B*和*Baichuan2-13B*的性能分别显著提高了14.22和19.71，超过了GPT-3.5的表现。这表明对模型进行适当的调整可以显著提高其性能。

**其他模型的局限性：**

*Llama2*、*ToolLlama*和*AgentLM*在规划、工具使用和反思方面表现不佳。这是因为这些模型无法为每个模板生成正确的格式，并且它们处理中文语境的能力有限。特别提到，*ToolLlama*在规划方面的表现比工具使用更差。研究发现在其训练数据中，“*thought*”字段经常被设置为“*none*”，这影响了规划指标的计算。这进一步强调了使用正确和一致的数据对模型训练的重要性。观察到反思比规划和工具使用更具挑战性，因为它需要模型具备更高的智力和更长的任务历史，以及更丰富的知识长度。这一点与前文提到的反思涉及更复杂的认知过程相呼应。

* 1. **人工评估**

首先，描述了一个评估代理系统的流程和评分标准。它强调了人类注释员在评估系统响应中的重要性和他们如何根据信息的准确性和完整性来为系统打分。此外，它也明确了什么是被视为“合格”的响应（即得分为4或5的响应）。

其次，关注于对三种不同的代理系统（*ReAct、AutoGPT*和*KAgentSys*）以及两种无代理模型（*Qwen-7Baichuan2-13B*）的性能评估。

它突出了以下几点：

(1). 代理系统通常比直接查询*LLM*产生更好的结果。

(2). *Qwen-7B*和*Baichuan2-13B*在没有代理的情况下也表现良好，这可能是由于它们在预训练或监督微调阶段集成了中文事实数据。

(3). *AutoGPT*的表现优于*ReAct，*因为其提示模板更复杂，且JSON输出格式更稳定。

(4). 无论使用哪种后端*LLM*，*KAgentSys*都取得了最好的结果。

(5). *GPT-4*与其他模型之间存在相当大的差距，开源模型的表现更差。但是，在经过元代理调整后，开源模型显示出显著的改进，其中*Qwen-7B*和*Baichuan-13B*的平均通过率和平均分数都有所增加。

(6). 经过元代理调整的最佳开源模型达到了74.13%的通过率，这比GPT-3.5高出10%，并接近*GPT-4*的83.58%。

* 1. **消融研究**

探讨了*KAgentSys*的性能改进和评估，包括对其关键组件的修改以及对其泛化能力的评估。以下是这段话的详细分析：

***KAgentSys*的关键组件修改：**简化了记忆机制，通过截断将任务、对话和外部知识插入提示中。用*Auto-GPT*中的基*本web\_search*和*web\_browse*模块替换了混合搜索-浏览工具集。这个简化版本被称为*KAgentSys-lite*，并且将被开源。

***KAgentSys-lite*的性能评估：**使用前面提到的人类评估方法以及*Baichuan2-MAT*作为认知核心来评估*KAgentSys-lite*的性能。结果显示,通过简化后的系统在通过率和平均分数上分别下降了5.97%和0.25。

***Meta-agent Tuning*方法在未见模板上的泛化评估：**排除了训练集中的所有开源代理模板，以评估我们的Meta-agent Tuning方法在未见模板上的泛化能力。训练后，Baichuan2-MAT在KAgentBench上的整体分数略有下降（45.34 → 42.12，下降3.22%）。这表明开源代理模板并不是我们训练过程中的关键组件，经过Meta-agent Tuning的模型在未见模板上表现稳健。

***KAgentLMs*对未见工具的泛化能力的评估：**在*KAgentBench*子集上进行了评估，重点关注训练集中不存在的功能。*Baichuan2-MAT*模型在计划指标上比*GPT-3.5-turbo*提高了15.71点（从21.34提高到37.05）。在工具使用指标上，*Baichuan2-MAT*比*GPT-3.5-turbo*提高了14.04点（从35.56提高到49.60）。

这些结果表明模型对未知工具的泛化能力稳健。

* 1. **个案研究(*case study*)**

首先，他们注意到*GPT-3.5*在生成规划阶段时存在重复性，即使在没有获取新信息或已经找到答案的情况下，它仍然采用相同的策略。相比之下，他们改进的方法展示了增强的反思能力和有效终止冗余过程的能力。

其次，在系统分析方面，传统搜索引擎在处理长尾或趋势问题时存在局限性。例如，如图3所示，当查询*Antonela*和*Messi*之间的年龄差异时，会出现两个问题：“*Messi*和他的妻子”这一查询的流行度使得搜索结果偏向于吸引用户关注的新闻文章，这些文章提供了与问题无关的内容，如关系时间线。这个问题涉及到他们的出生日期（一个不常见的细节），是一个“长尾”搜索查询。大型语言模型（*LLMs*）单独处理这个问题有困难，因为它们可能只记得*Messi*的出生日期而忘记了*Antonela*的。

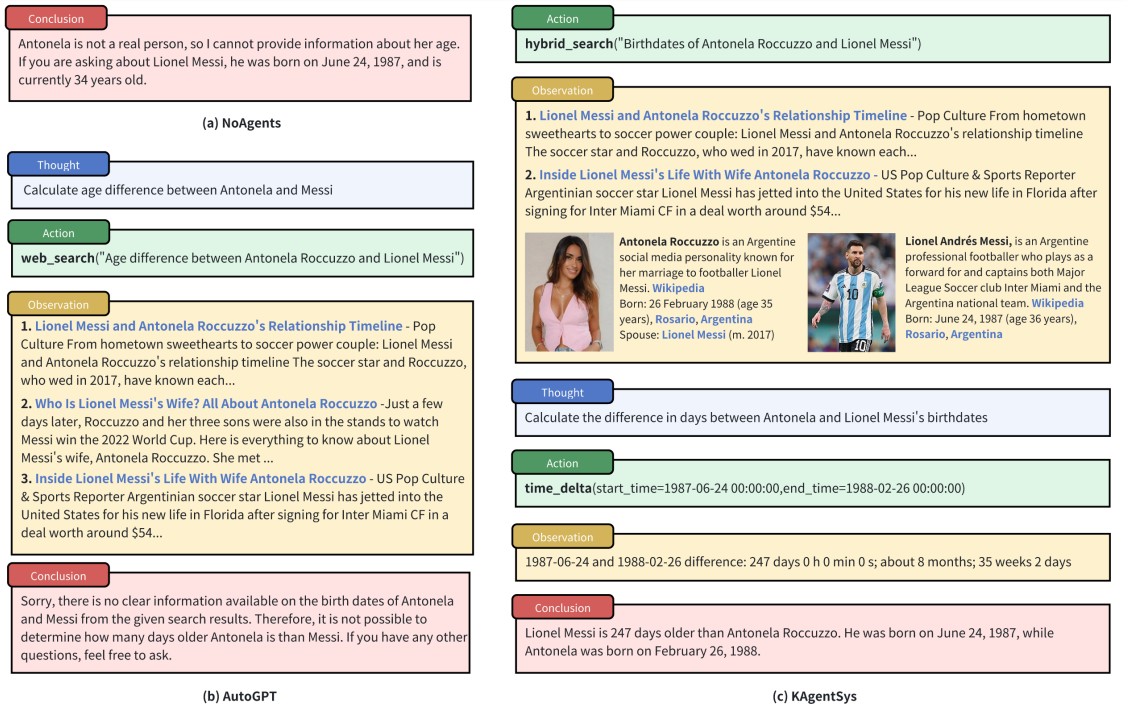


图3：当回答“安东内拉比梅西大多少天”的问题时，不同系统的决策路径。

将*LLMs*与搜索引擎相结合也无法解决问题，因为所提供的信息虽然相关，但无法精确地回答查询。他们的方法通过结合实体链接和从资源（如*Wikipedia*）中提取相关信息来克服这些障碍。具体来说，他们的系统首先检索*Messi*和他妻子的出生日期，然后使用*time\_delta*工具准确地计算时间差，从而对提出的问题给出正确的答案。