GOFAI 遇见生成式 AI：通过大规模语言模型开发专家系统

Eduardo C. Garrido-Merchán[1,2] and Cristina Puente[3]

[1]*Quantitative Methods Department, Comillas Pontifical University*
[2]*Institute of Research in Technology (IIT), Madrid, Spain.*
[3]*Computer Science Department, ICAI School of Engineering, Comillas Pontifical University, 28015 Madrid, Spain*

2025 年 7 月

摘要

大型语言模型（LLMs）的发展成功地变革了基于知识的系统，如开放领域的问题回答系统，能够自动生成大量的看似连贯的信息。然而，这些模型存在一些缺点，比如幻觉现象或自信生成错误或无法验证的事实。本文介绍了一种利用 LLMs 以受控和透明的方式开发专家系统的全新方法。通过限定领域并采用结构良好的基于提示的提取方法，我们生成了 Prolog 中的知识符号表示形式，这种形式可以由人类专家进行验证和修正。这种方法还保证了所开发专家系统的可解释性、扩展性和可靠性。通过对 Claude Sonnet 3.7 和 GPT-4.1 进行定量和定性实验，我们展示了生成的知识库在事实准确性和语义连贯性方面具有很强的一致性。本文提出了一种透明的混合解决方案，将 LLMs 的记忆能力与符号系统的精度相结合，为敏感领域的可靠 AI 应用奠定了基础。

1 介绍

自 70 年代和 80 年代开发出首批专家系统如 DENDRAL[8, 10], MYCIN[29], XCON[2] 及其他一些系统以来，这项技术作为知识与建议的辅助手段，在诸如医疗、法律、教学等众多任务中不断演进。

创建专家系统时面临的最大复杂性之一是为其提供完整且可靠的知识。为此任务，通常会使用一位对该主题有深入了解的人类专家以及大量问卷或其他方法来捕捉以图、决策树、知识规则 [13] 等形式获得的知识，以便尽可能有效地提取所获知识 [23]。

随着 2010 年大型语言模型（LLM）的出现，拥有海量数据信息容量的系统，在生成针对特定任务的小领域系统方面开启了一个全新的世界。因此，医疗、教育和完成类的聊天机器人迅速被创建出来，允许流畅对话并回答多种新的问题 [9]。问题是，并非系统返回的所有知识都是可靠的。正如 [15] 和 [18] 所指出的那样，LLM 在急于给出答案时会产生幻觉，即错误或不准确的知识作为答案。这里的问题很严重，因为如果知识来源不可靠，答案也不应可靠，因此，除非我们进行检查，否则系统将传播错误信息。

大规模语言模型中的幻觉是系统在数据不准确或内部故障时产生的虚假响应。在将大规模语言模型应用于系统中时，检测这些幻觉至关重要，因为这可能会导致医疗、科学、商业等敏感过程中出现严重错误，并且由于缺乏验证或解释 [24]，可能导致网络中虚假信息的传播。

幻觉的产生可以归类为以下原因中的失败：[27, 20]：数据失败 [7]，因为它们不是最新的、存在偏见的数据或与假新闻混合在一起，访问某些信息源时存在知识产权问题等。在图 1 提出的情况下，有

明显的过时信息案例，因为在提问前的 15 天内系统数据中没有教宗弗朗西斯去世的信息。即便如此，它仍然断言截至 5 月 7 日，教宗仍然健在。为了解决这个问题，建议使用可靠的信息来源，对比信息（例如，在医疗问题的情况下，梅奥诊所网站上有大量的信息 `https://www.mayoclinic.org/es`），并且如果信息不可获取或没有足够的信息来形成一个恰当的回答，则不要回答该问题。

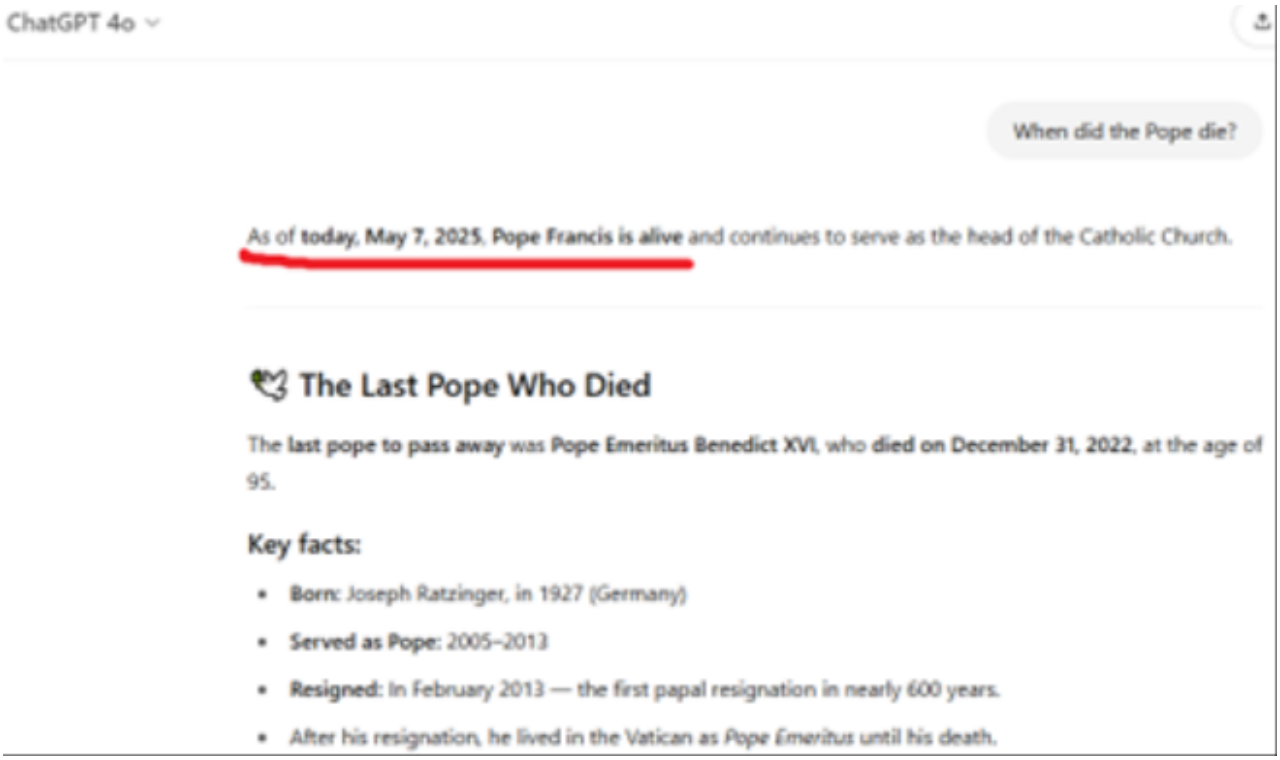

图 1: ChatGPT 的回答错误地确认了教宗弗朗西斯科在 2025 年 5 月 7 日还活着，而事实上他于 2025 年 4 月 21 日去世。

模型和训练失败，架构问题，RNN 与 Transformer 的比较，理解问题上下文和复杂性的困难，在数据训练中生成答案时出现过拟合问题。

而且，生成答案时会出现问题，因为很多时候大语言模型更注重呈现文本的流畅性而不是信息的准确性。较高的采样温度会增加随机性，而专注于概率最高的词语可能会导致文本质量下降 [26]。

有几种提议来解决这些问题，包括使用可靠的外部来源，在应用程序本身内部进行事实核查以确保显示相同的结果（使用同义词、不同的温度等），或者训练系统检测错误答案 [14, 25]。

为了提出解决此问题的另一种方案，我们设计了一个系统来处理一个更小、更具体的知识领域。通过一组预定义提示 [16, 1, 19, 12]，像 Chatgpt 这样的大语言模型可以生成一个可管理的知识库，并将其转化为易于验证的模型（例如规则系统 [30]），由该领域的专家进行转换。通过这种方式的验证，我们在生成专家系统时保证了三个非常重要的优势：

- 系统的可解释性。
- 更大的信息量（因为以这种方式生成信息比通过专家更容易）。
- 信息的真实性和可靠性（因为如果专家发现所呈现的模型中存在任何错误，将立即进行纠正）。

关于专家系统的创建，Prolog 是一种基于一阶谓词逻辑的著名逻辑编程语言，由于其声明式模式匹配语法和推理机制，在开发专家系统方面长期受到欢迎。系统知识通过事实和规则表达，并使用反向链接和解析来推断新知识。著名的应用包括医疗诊断系统（例如 MYCIN）、法律推理工具 [3] 以及

复杂产品的配置 [28, 5]。例如，Prolog 是临床问答系统 CLINIQA[22] 和 XCON（也称为 R1）所使用的编程语言，后者是 Digital Equipment Corporation 使用的一种基于规则的系统，用于指定如何配置机器 [21]。其处理符号推理的能力使其成为构建辅导系统和智能顾问 [11, 6] 的合适工具。对于需要解释性、透明性和基于规则的推理领域的专家系统编程来说，Prolog 仍然是一个非常好的选择。

为了实现这些目标，我们建立了一个模型，通过一组特定的提示（由专家设计）查询 LLM 中的信息，然后生成一个逻辑规则模型以简化专家验证阶段的工作，提出一种校正方法来清除来自不精确、错误数据或数据偏见的信息，并创建一个可靠的专家系统。因此，本文分为以下几个部分。首先，我们从介绍自 20 世纪 70 年代以来专家系统的演变以及大型语言模型（LLMs）在知识抽取的新时代中的作用开始，这些系统如聊天机器人和推荐引擎。我们指出，LLMs 倾向于产生幻觉、生成错误或无法验证的输出，这可能在医疗、教育、法律等重要任务中导致有害决策。

预备知识和相关工作部分，介绍了为本项目奠定基础的最新研究成果，回顾了其成就和主要贡献。在方法论部分，我们开发了一种协议来选择、定义和处理大型语言模型的知识。对于此任务，我们使用了定制化的提示来从语言模型中提取结构化信息。这些信息以 Prolog 表示，它在透明性、可解释性和基于规则的推理方面表现出色。在方法论部分，我们介绍了一个递归过程，该过程通过询问 LLM 构建概念图，并将 LLM 的回答翻译成 Prolog 事实和关系，同时验证知识的语法和语义正确性。实验部分展示了定量和定性的结果。我们还证明了生成的 Prolog 专家系统可以成功执行并查询而不会出错，确认其实际应用的可行性。最后，在最后一部分中，提出了结论和未来工作。

2 初步和相关工作

在本节中，我们将回顾在这个领域开发并作为我们系统基础的工作。我们将它们分为三个主要部分。一方面，信息收集，我们将讨论查询 LLM 信息所使用的方法。其次，我们将重点放在知识的形式化表示及其验证上，最后我们将讨论专家系统的构建。

自 2010 年以来，随着 LLMs、LlaMA、Gpt3、Copilot 等大型语言模型的出现，依赖这些系统的应用程序层出不穷，从推荐系统、聊天机器人、分类器、自动程序员等，彻底改变了数字世界。

2.1 知识获取

知识生成经历了一种与传统方法不同的算法处理，例如微调 [19] 或通过提示从大语言模型本身提取知识（[17]）。这种方法通常用于将大语言模型的知识领域缩小为一个更具体、可控制和透明的领域。

对于这项任务，并基于这些工作，我们设计了一整套提示，涵盖了我们希望在这些实验中测试的知识领域。

2.2 关于验证系统和可计算性的大语言模型的局限性

尽管大型语言模型（LLMs）在从自然语言生成到代码补全等各种任务上具备惊人的能力，其表现依然鲜明地划定了经典可计算性理论所设定的界限。例如，当被要求检查任意一段代码是否会终止时，LLMs 有时会给出看似有说服力的理由或甚至权威性的答案。然而，这并不是由于任何决策属性的结果，而是训练数据中的统计推断所致。图灵证明了停机问题是不可判定的，即不存在能够决定所有可能程序是否终止的算法——即使是在 LLMs 中隐式编码的也不是（[4]）。因此，模型做出的确信预测只是对理解的模拟而非实际验证。类似地，当被要求在形式逻辑中证明某个命题或进行逐步代数操作时，LLMs 经常会虚构中间步骤或错误应用逻辑原则，展示了它们缺乏正式的基础。

这是知识差距在正确性不仅仅依赖于可能性，而是证明的领域中特别具有共鸣的地方，例如形式方法、密码协议验证、定理证明。

困难不是规模问题，而是原则问题：LLMs 受限于丘奇-图灵论题，完全工作在可计算函数的宇宙内，因此共享所有足够复杂的形式系统所受的不可判定性和不完全性定理。

相应地，LLMs 提供的是一个用于启发式探索的良好概率阴影，但原则上无法提供保证。

这样，LLMs 无所不知的幻觉可能会掩盖那些在 AI 时代与哥德尔和图灵时代一样相关的理论边界。

3 方法论

本节介绍了从大型语言模型（LLM）中提取结构化符号知识并将其编码到基于 Prolog 的专家系统的框架方法。目标是构建一个形式化的、逻辑一致的知识库，以表示围绕用户定义概念的语义景观。所提出的系统结合了递归提示链接、与本体对齐的关系抽取以及一阶逻辑中的符号编码。

管道是用 Python 实现的，并通过接收根关键字 $T \in K$ 的 shell 接口调用，其中 K 表示所有概念域的集合。知识获取过程由两个超参数驱动：水平广度 $h \in N$，它限制每个节点检索到的相关语义概念的数量，以及垂直深度 $d \in N$，它定义了以 T 为根的概念图的最大深度。系统通过结构化提示与大语言模型互动，接收包含概念节点和标记的语义关系的 JSON 格式响应，这些信息使用受控谓词词汇表被转换成 Prolog 事实。

形式上，输出是一个有向标记图 $G = (C, R)$，其中：

- $C \subseteq K$ 是提取的概念集合。
- $R \subseteq C \times L \times C$ 是概念之间关系的集合。
- L 是一组有限的谓词标签，包括通用形式（相关于、意味着、导致、与……有关）和领域特定的扩展。

为了管理图形扩展，我们定义了一个递归语义扩展算子：$\Psi : C \longrightarrow \mathcal{P}(C \times L \times C)$ 该算子将一个概念映射到最多 h 个标记关系的集合，并沿着新发现的节点递归应用直到深度 d。遍历是广度优先的，通过记忆已访问的概念来避免循环。我们在算法 1 中总结了这些步骤。

Algorithm 1: 递归语义扩展与符号编码

输入: T：字符串 // 根概念
h：整数 ≥ 1 // 水平宽度
d：整数 ≥ 0 // 垂直深度

输出: K_T：Prolog 文件 // Prolog 知识库

procedure BUILD_KNOWLEDGE_BASE(T, h, d)
  $Q \leftarrow [(T, 0)]$ ▷ Queue of (concept, depth)
  $V \leftarrow \emptyset$ ▷ Visited concepts
  $F \leftarrow \emptyset$ ▷ Set of unique fact hashes
  $KB \leftarrow \emptyset$ ▷ Prolog fact accumulator
  ;
  而 $Q \neq \emptyset$
    $(c, \text{depth}) \leftarrow Q.\text{dequeue}()$
    如果 $c \in V$ **或者** $\text{depth} > d$
      继续
    结束_if
    $V \leftarrow V \cup \{c\}$
    $\text{Json} \leftarrow \text{Query\_LLM}(c, h)$
    $(\text{Concepts}, \text{Relations}) \leftarrow \text{Parse\_JSON}(\text{Json})$
    对于所有 $c' \in \text{Concepts}$
      如果 $c' \notin V$
        $Q.\text{enqueue}((c', \text{depth} + 1))$
      结束如果
      $KB \leftarrow KB \cup \{\text{concept}(c'), \text{related\_to}(c', c)\}$
    结束循环
    对于所有 $(s, r, t, \text{explanation}) \in \text{Relations}$
      $KB \leftarrow KB \cup \{r(s, t)\}$
      存储_解释 $(r(s, t), \text{explanation})$
    结束循环
    去重 (KB, F)
  ;
  结束循环
  编码_知识库 $(KB) \rightarrow K_T$
  验证_前序 (K_T)
  return K_T
end procedure

每个提取的概念 c 被编码为一个一元谓词概念 (c)，每个语义关系 $(c1, r, c2)$ 被编码为一个二元谓词 $r(c1, c2)$，其中 $r \in L$。由大语言模型生成的自然语言解释被保留为内联 Prolog 注释，使用 %Explanation: 指令。系统通过事实的词汇规范化和哈希来强制唯一性，并在部署前使用 SWI-Prolog 进行语法验证。

该算法支持以任意输入主题为基础构建多层概念图，能够在可解释且可扩展的专家系统框架内实现递归推理和逻辑查询。

该系统支持通用关系抽取和领域专用关系抽取，能够构建粒度和范围各异的知识图谱。在其默认配置中，模型会根据输入主题的分类领域（如哲学、历史或文学）来调整提示语，从而生成如 developed_by、lived_in、wrote、pioneered 或 invented 这样的关系，这些关系捕捉了传记信息、地理信息和作者信息。这使得知识表示更加丰富且具有学科特定性。然而，该系统可以配置为最小模式运行，将本体限制为核心谓词概念 concept/1、related_to/2、implies/2 和 causes/2，从而专注于抽象的概念结构和逻辑因果关系。这种模块化设计使灵活部署成为可能，适用于需要高可解释性或轻量级逻辑推理的应用程序。我们在表 1 和表 2 中描述了所有谓词。

表 1: 系统支持的核心谓词。

Category	Predicate	Arity	Description
Core (Causal)	concept/1	1	Declares a concept
	related_to/2	2	Links a concept to another
	implies/2	2	Logical implication between concepts
	causes/2	2	Causal relation between concepts
	relates_to/2	2	Generic semantic relation

作为最后一个组件，我们开发了一个可视化工具，该工具可以自动解析生成的 Prolog 知识库，并将其概念结构呈现为有向图。该可视化器使用颜色编码的边来区分核心因果谓词（例如，causes/2, implies/2, related_to/2）和领域特定的关系（例如，written_by/2, developed_by/2），并输出为带标签的 PDF 图形。这有助于快速检查知识拓扑结构，支持定性验证，并在跨学科背景下增强可解释性。我们在图 2 和 3 中包含了一些示例。

在两种情况下，图 2 和图 3 中，用户运行一个包含所有系统配置的 shell 脚本，该 shell 脚本将其发送给一个通过 LLM API 与 LLMs 通信的知识提取过程。然后，LLM 将专家系统构建者所需的知识返回，并持久化到 Prolog 文件中。最后，用户可以在专家系统的 Prolog 文件中验证 LLM 的信息。我们在图 4 中总结了方法论部分所有逻辑的流程图。

4 实验部分

提出的系统通过两个互补的实验进行了评估：(i) 一项定量知识验证研究，测量事实准确性与已知参考标准进行对比；(ii) 定性案例分析，评估算法在多个语义扩展水平上的结构行为，包括图可视化。这种双重设计的目标是评估生成的知识库的本体有效性以及结构表达力。我们选择了 Claude Sonnet 3.7 和 GPT 4.1 进行这些实验，因为它们是 LMArena 中排名前十的大型语言模型样本之一。我们认为进行这些实验足以说明算法的行为。

4.1 事实验证通过统计假设检验

为了评估从大语言模型中提取的符号知识是否反映了可验证的事实准确性，我们设计了一种使用精心挑选的参考事实集进行统计评估的方法。总共随机选择了 n=25 个查询，这些查询跨越了三个具有客观事实基础的内容领域：历史、文学和哲学。每个查询都被配置为生成包含近 300 行 Prolog 事实和关系的知识库。从这些事实和关系中我们抽取了 10 行随机知识，总计得到了 250 条断言用于系统评估。这些事实被手动与已确立的知识来源（百科全书和学术）进行了对比，并标记为正确或错误。

我们定义零假设 H0：提取知识的真实准确率小于或等于 80%（即 $p \leq 0.80$）。备择假设 H1：真实准确率超过 80%（即 $p > 0.80$），其中 p 是查询中事实准确性的人口平均值。我们应用了一侧 t 检验，

表 2: 系统支持的特定领域谓词。

Category	Predicate	Arity	Description
Philosophy	response_to/2	2	Theory formulated as response to another
	school_of_thought/2	2	Philosopher belongs to school
	main_work/2	2	Main philosophical work of thinker
	lived_during/2	2	Philosopher lived during period
	developed_by/2	2	Concept developed by philosopher
	influenced_by/2	2	Philosopher/work influenced by another
	criticized_by/2	2	Theory criticized by philosopher
Literature	written_by/2	2	Work written by author
	published_in/2	2	Work published in year
	set_in/2	2	Setting or period of literary work
	protagonist_of/2	2	Character is protagonist of work
	genre_of/2	2	Genre of a work
	movement/2	2	Work or author belongs to movement
	influenced_by/2	2	Work influenced by another work
	adapted_into/2	2	Source work adapted into another form
Arts	created_by/2	2	Artwork created by artist
	created_in/2	2	Year or period of creation
	belongs_to/2	2	Artist/work belongs to movement/style
	housed_in/2	2	Artwork housed in location
	technique_used/2	2	Artistic technique applied
	commissioned_by/2	2	Work commissioned by patron
	trained_under/2	2	Artist trained under another
	influenced_by/2	2	Artistic influence from earlier artist
History	born_in/2	2	Person born in year/place
	died_in/2	2	Person died in year/place
	occurred_in/2	2	Event occurred in time period
	located_in/2	2	Entity located in place
	preceded/2	2	One event/entity precedes another
	succeeded/2	2	One event/entity succeeds another
	founded_by/2	2	Institution founded by person
	ruled_during/2	2	Ruler ruled during period
	contemporary_of/2	2	Temporal coexistence between people

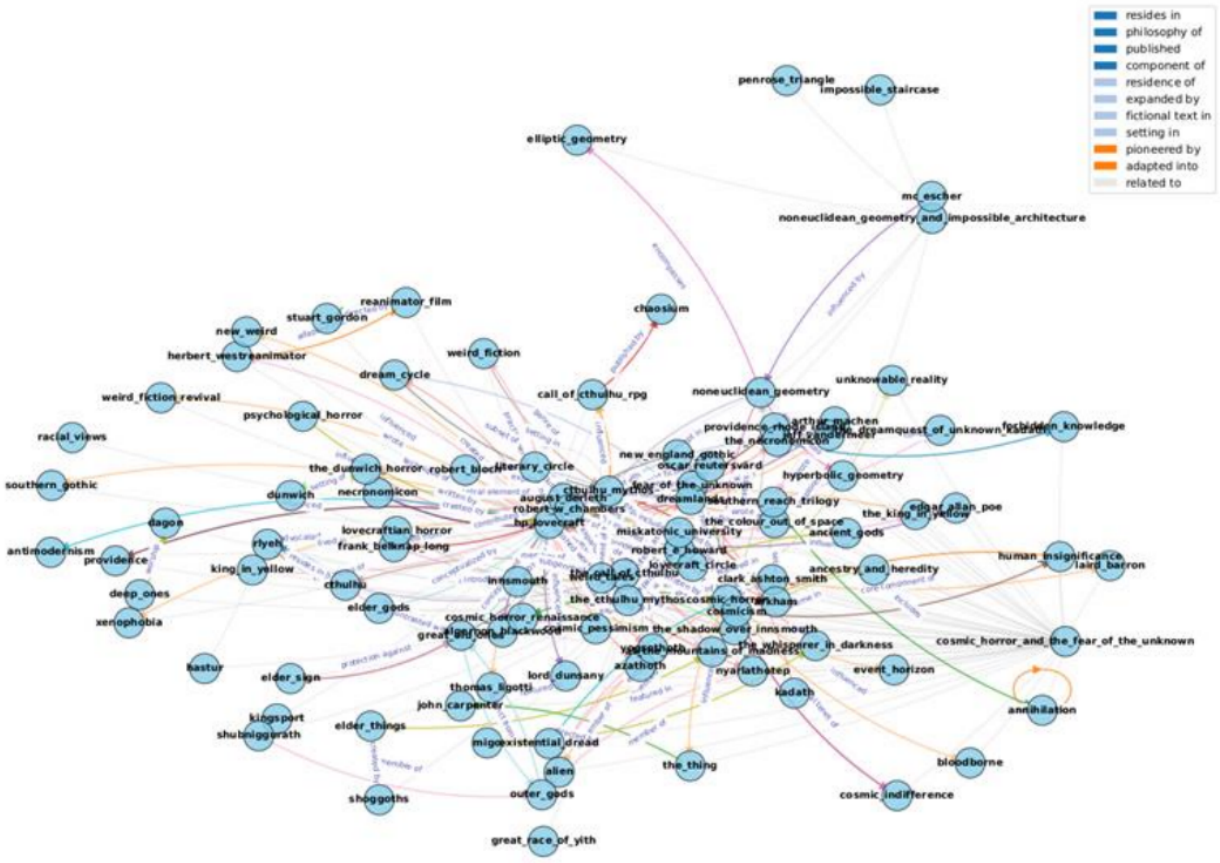

图 2: 关于宇宙恐怖小说作家 H. P. Lovecraft 的因果关系和领域关系从 Claude 3.7 Sonnet 中提取出的 100 个节点的知识可视化。

将样本均值比例与基准阈值 0.80 进行比较。统计显著性在 $\alpha = 0.05$ 处进行了评估，因为这是此类研究的常见值。得出的 p 值和置信区间使我们能够确定观察到的事实一致性是否在统计上高于通常期望的领域适应型 LLMs 参考准确率阈值。

此测试作为语义保真度的定量代理，衡量从 LLM 提取的符号结构是否保留了足够的真实性以被视为专家可用。作为查询提交给 LLM 的随机主题列于表 2 中。

正如我们所说，从这些主题中我们随机选择了知识库中的 10 行，并在学术来源中评估内容是否真实。所有提取的知识都存储在一个 pl 文件中，并在这些来源中进行评估。评估考虑信息是否准确以及句子是否正确，因为有些句子是 Prolog 工具。独立评估的结果是只有错误的事实是 Thomas Aquinas 的一个句子的年份构建不佳和一个递归关系。由于样本量为 250，我们可以执行单一样本比例 z 检验。

对于 Claude Sonnet 3.7，所获得的 p 值接近零，为 1.59872e-14，因此我们可以安全地拒绝知识提取不准确的假设，并接受 Claude Sonnet 3.7 蒸馏到专家系统至少达到 80% 准确性的假设。我们还可以从频率主义的角度根据结果、样本大小和 5% 的置信度构建准确性比例的信心，此时的准确性为 0.992 ± 0.01104 使用标准推断。从贝叶斯视角出发，假设所有谓词都是独立采样自提示 LLM 的条件分布，我们可以论证检索的准确性是正态分布的。因此，我们将获得的准确率 99.2% 用作均值的点估计，但是作为超先验我们考虑其中的一项事实可以被认为是标准差，这虽然非常悲观但在我们的信念下是合理的。因此，这些特定主题的准确性可以用 $N(0.992, 1/250)$ 来建模。进一步的应用可以将这个 80% 的阈值提高到所需水平，并且在人工修正之后它可以增加直到给定样本的 100%。

对于 GPT 4.1，应用了相同的算法，得到了类似的结果。这次的准确率为 0.996，关联的 p 值更小为 4.88498e-15，因为只有一个句子是错误的。因此，在这里我们以 95%的信心水平拒绝了原假设。使

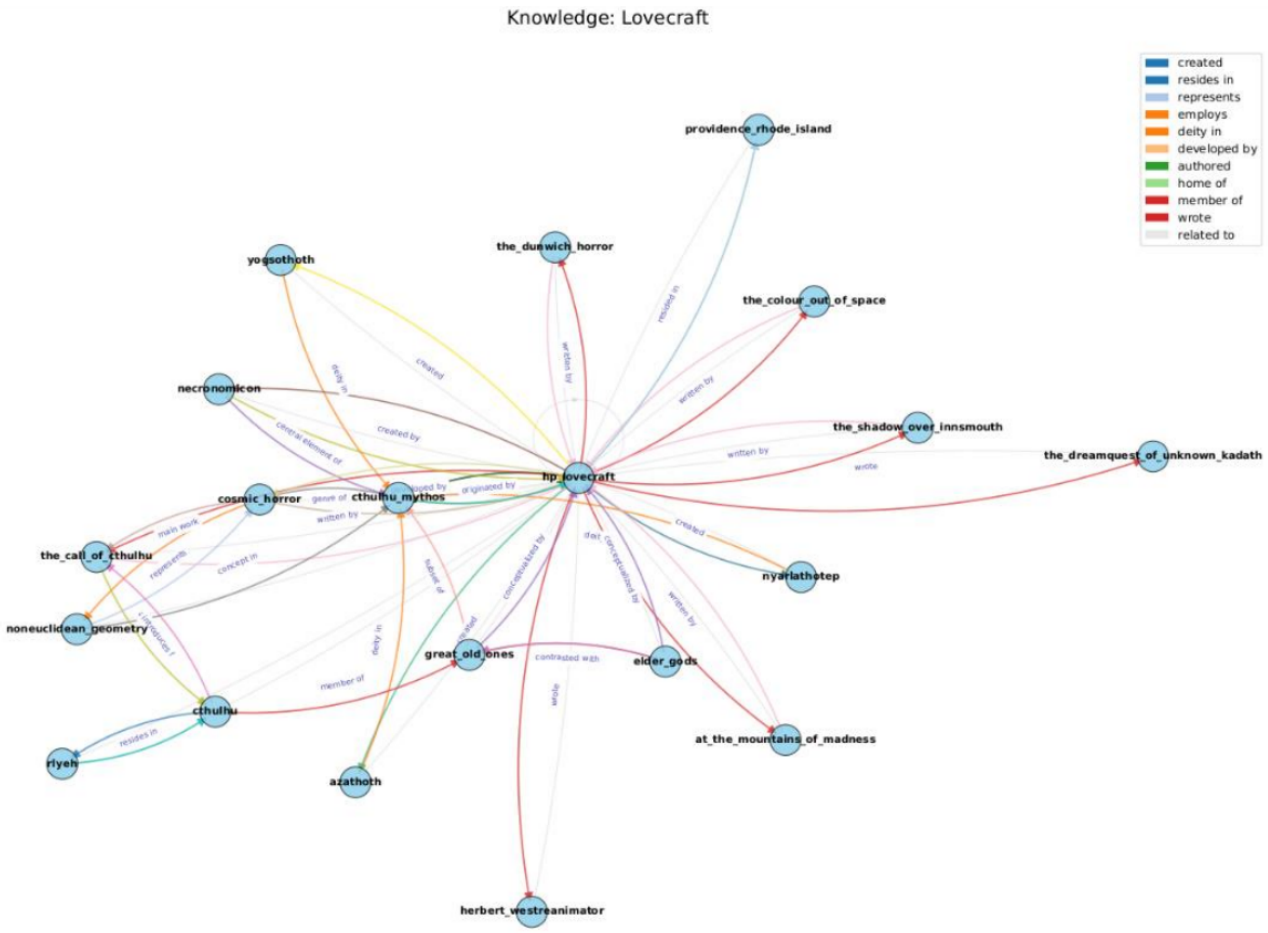

图 3: 节点可以使用可视化脚本进行修剪。我们在图中包含了与前一个图相比具有相同概念的 20 个因果和领域关系的节点

用标准推理，比例为 0.996 ± 0.007824，并且上限被限定在 1，这与 Sonnet 的情况类似。按照之前的贝叶斯推理逻辑，我们可以将先验设定为右截断高斯分布 N(0.996, 0.5/250)。最后，我们还可以执行统计假设检验过程来评估两个模型所获得的差异是否显著，原假设是两种比例相似而备择假设是在相同信心水平下（5%）它们不同，在两个总体中样本大小类似。正如预期的那样，我们得到了一个较高的 p 值，为 0.5625，这表明没有足够的实证证据来拒绝原假设。因此，我们可以看到在一个验证系统中，两种系统所表示的知识准确性在统计上是相似的，这是其他前 10 名 LMArena LLM 模型也可能表现类似且我们可以通过我们的算法提取它们的知识以构建专家系统的实证证据。

我们建议使用这种方法论来验证关于某个主题或一系列主题的大语言模型的知识。本节和研究问题的目的不是评估 Claude Sonnet 3.7，而是展示一种评估来自任何大语言模型的知识并将其存储在知识数据库中的方法。将知识数据库编码到专家系统中的优势在于，专家可以纠正系统中的幻觉，并且对系统的查询是透明的、可解释的、可理解的，可以使用逻辑推理引擎推断新知识，并且结果是确定性的。总之，通过这种方法论，我们可以结合两个世界的最佳之处。大语言模型的记忆力和符号系统的精确性。

4.2 不同展开深度的定性行为

除了事实准确性之外，我们对系统在多个语义扩展深度（d=0,1,2,3）和固定水平广度（h=30）上的输出进行了定性和结构评估。使用根主题 “Plato”，我们生成了一系列知识库，用于增加 d 的值，每

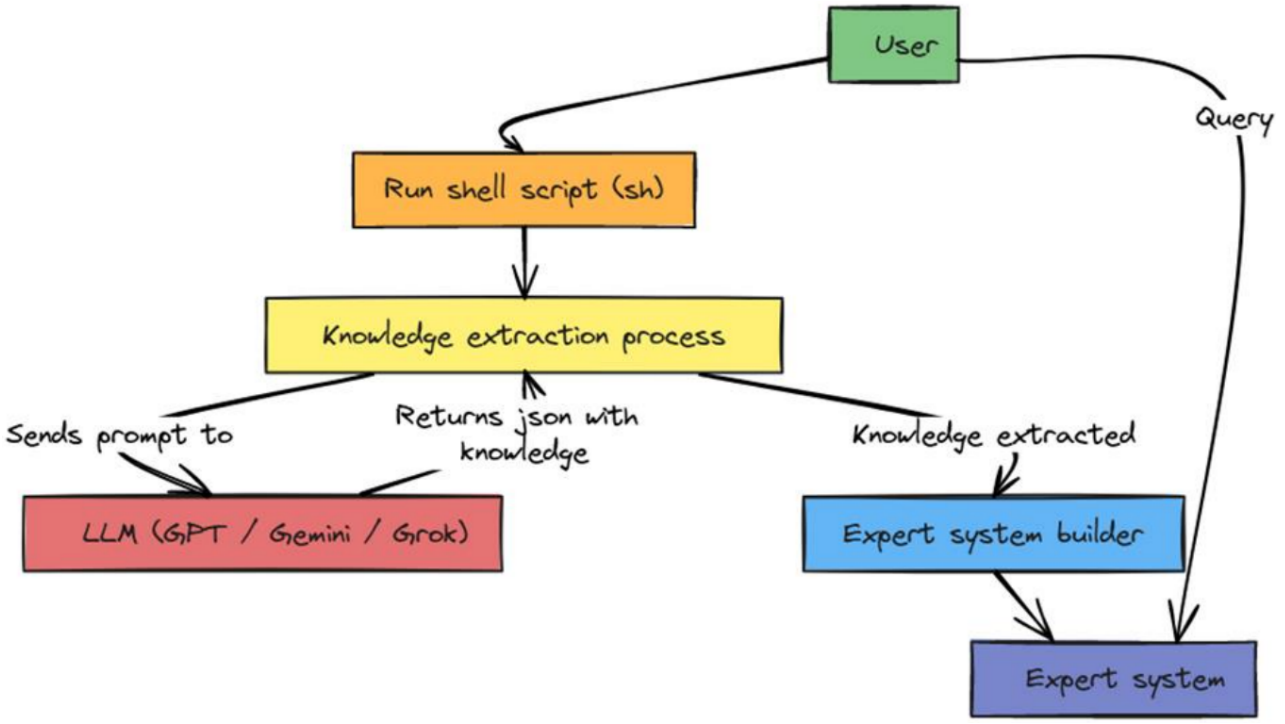

图 4: 包含从大语言模型创建专家系统所涉及的所有流程的示意图。

个知识库都使用 Claude Sonnet 3.7 捕获更广泛的概念邻域。对于每个深度级别，我们记录了提取的概念和关系的数量、谓词类型的多样性以及出现的语义集群。该系统可以简单地被调用以满足该过程：

```
python3 claude_to_Prolog.py "Plato" --depth 3 --max-topics 30 --output
"plato_knowledge_network.pl" -report
```

生成的 .pl 文件使用第 3 节中描述的专业图形可视化器进行了解析和可视化。结果图形证实了随着 d 的增加概念边界的扩展，揭示了一致的子结构，例如柏拉图理论的本体（例如 theory_of_forms, divided_line）、传记数据（lived_during, main_work）以及影响模式（influenced_by, criticized_by）。这些可视化既用作图形一致性的诊断工具，也作为系统生成的符号结构可解释性的定性展示。包含专家系统的所有代码将随文章被接受后上传到 Github。我们在图 5 中可视化了柏拉图的 10 种主要关系。

我们为了说明目的重复了相同的流程，使用 Grok 3 查询关于柏拉图的内容，通过与 Claude Sonnet 3.7 相同的过程获得了以下图形，如图 6 所示。

有趣的是，我们看到有些概念是相似的，比如哲学王、三部分灵魂或洞穴比喻。然而，在 Grok 中我们发现了一些不同的概念，如柏拉图式的爱情 eros。总体而言，定量和结构分析支持这样的结论：所提出的方法产生了逻辑连贯、事实依据充分且语义表达丰富的知识表示形式，适用于专家系统部署或可解释的人工智能应用。

我们进行了一项最终实验来测试通过此程序生成的专家系统是否能够从 Prolog 推理引擎中执行和查询。具体来说，我们为这项任务选择了 SWI-Prolog 版本 9.2.7 for x86_64-linux 引擎，并从 GPT 4.1 中提取知识。可以通过修改代码将此过程轻松适应其他引擎，如 Ciao Prolog。

之前类别中选择的随机主题是这些：简 · 奥斯汀，加布里埃尔 · 加西亚 · 马尔克斯，威廉 · 莎士比亚，马克 · 吐温，玛雅 · 安杰洛，村上春树，陀思妥耶夫斯基，夏洛特 · 勃朗特，西尔维娅 · 普拉 ，詹姆斯 · 乔伊斯，勒内 · 笛卡尔，卡尔 · 马克思，西蒙娜 · 德 · 波伏娃，米歇尔 · 福柯，约翰 · 洛克，托马斯 · 霍布斯，希坡的奥古斯丁，让-雅克 · 卢梭，路德维希 · 维特根斯坦，玛丽 · 沃斯通克拉夫特，《黑死病》，《十字军东征》，第二次世界大战，《新教改革》和《太空竞赛》。我们忽略了文件中子句不

表 3: 随机选择的主题用于评估验证系统

主题	**域**
H.P. Lovecraft	Literature
Søren Kierkegaard	Philosophy
The French Revolution	History
Franz Kafka	Literature
Plato	Philosophy
World War I	History
Virginia Woolf	Literature
Immanuel Kant	Philosophy
The Cold War	History
Dante Alighieri	Literature
Aristotle	Philosophy
The Renaissance	History
Homer	Literature
Jean-Paul Sartre	Philosophy
The Industrial Revolution	History
Emily Dickinson	Literature
Friedrich Nietzsche	Philosophy
The American Civil War	History
Leo Tolstoy	Literature
Thomas Aquinas	Philosophy
The Fall of the Roman Empire	History
Mary Shelley	Literature
David Hume	Philosophy
The Age of Exploration	History
Jorge Luis Borges	Literature

连续的警告，因为这些警告可以通过执行命令或在.pl 文件中包含特定的不连续指令来轻松抑制。我们只是对生成的文件进行了一个查询子句以及 “concept(X), !.” 谓词，以测试是否一切都能无误地被执行。我们确认所有生成的 pl 文件都已读取并查询且没有错误。因此，我们可以确认该过程对于不同的主题和通过 API 执行 LLM 的多次运行是稳健的。

5 结论与未来工作

在本文中，我们提出了一种结合大型语言模型的结构化知识提取和通过 Prolog 符号编码进行人工验证的混合方法。该方法解决了开发大型语言模型时面临的一个主要问题：其输出中的幻觉和不可验证性。通过对知识领域进行约束、使用精心设计的提示，并将提取的信息编码为基于逻辑的形式，我们能够创建准确、可解释且可重复使用的知识。

定量来看，我们的方法对于从 Claude Sonnet 3.7 和 GPT-4 这样的模型中提取的事实实现了超过 99%的准确性，并通过统计确认否定了准确率低于 80%的零假设。此外，定性分析表明该系统生成了

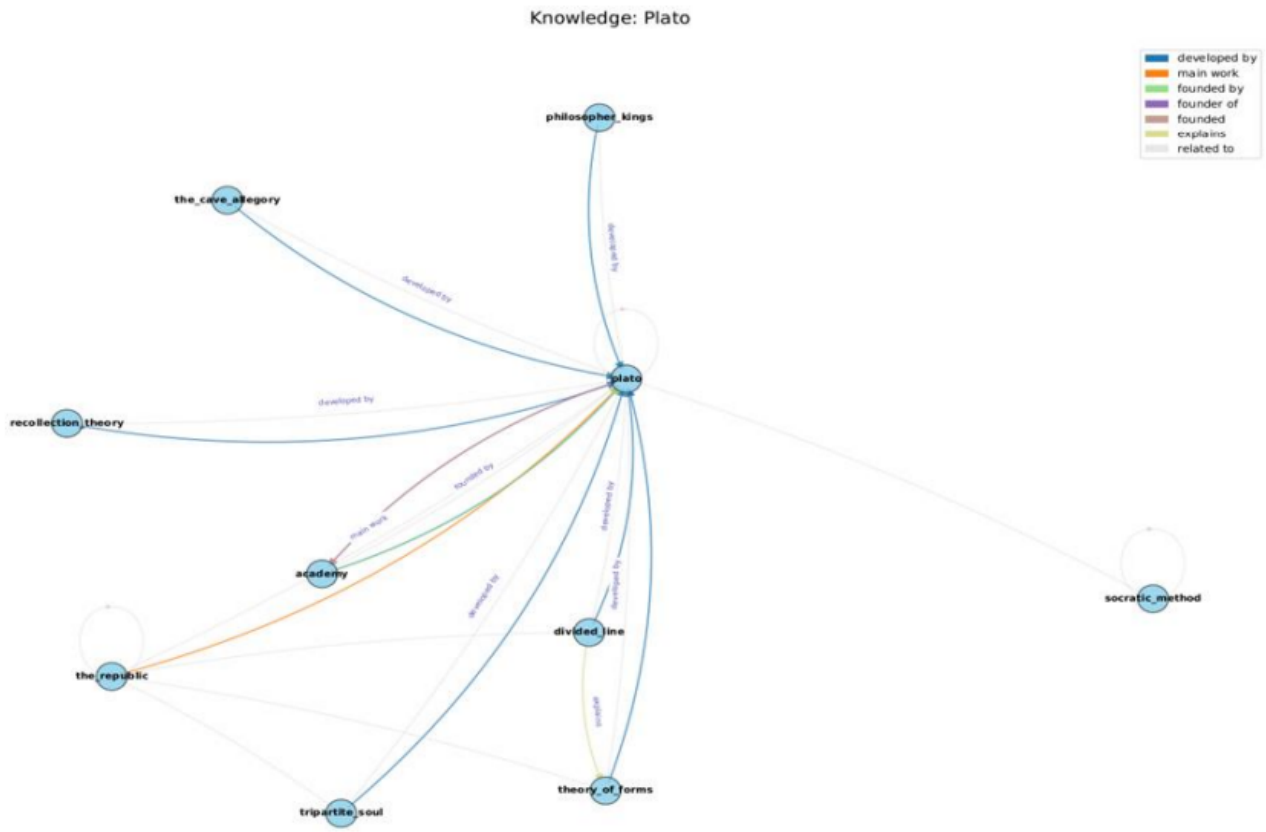

图 5: 柏拉图的大型语言模型知识网络。10 个主要关系。我们可以看到柏拉图发展的著名概念，如洞穴寓言、哲学王、理想国、理念论或三分灵魂理论，或者柏拉图创立了学园。

有意义的语义扩展和结构知识图谱，展示了其在多层次推理和概念表示中的应用。

此类技术不仅能够构建更可靠的专家系统，还提供了一种实用且可扩展的方法来验证 LLM 生成的知识在医学、教育或法律等关键领域的有效性。符号知识库允许人类纠正幻觉，进行确定性的逻辑推理并控制推理过程的透明度，这些都是纯粹的统计方法所缺乏的特性。

对于未来的行，本实验留下了一条研究路线，以量化每个大语言模型提供的新信息量以及与其他大语言模型提取的信息相比的质量，可测量为每个多语言模型关于新主题的熵减少量。

参考文献

[1] Alkhulaifi, A., Alsahli, F., and Ahmad, I. Knowledge distillation in deep learning and its applications. *PeerJ Computer Science 7* (2021), e474.

[2] Barker, V. E., O'Connor, D. E., Bachant, J., and Soloway, E. Expert systems for configuration at digital: Xcon and beyond. *Communications of the ACM 32*, 3 (1989), 298–318.

[3] Borrelli, M. A. Prolog and the law: Using expert systems to perform legal analysis in the uk. *Software LJ 3* (1989), 687.

[4] Boyer, R. S., and Moore, J. S. A mechanical proof of the unsolvability of the halting problem. *Journal of the ACM (JACM) 31*, 3 (1984), 441–458.

[5] Bratko, I. *Prolog Programming for Artificial Intelligence*, 4th ed. Pearson Education, 2012.

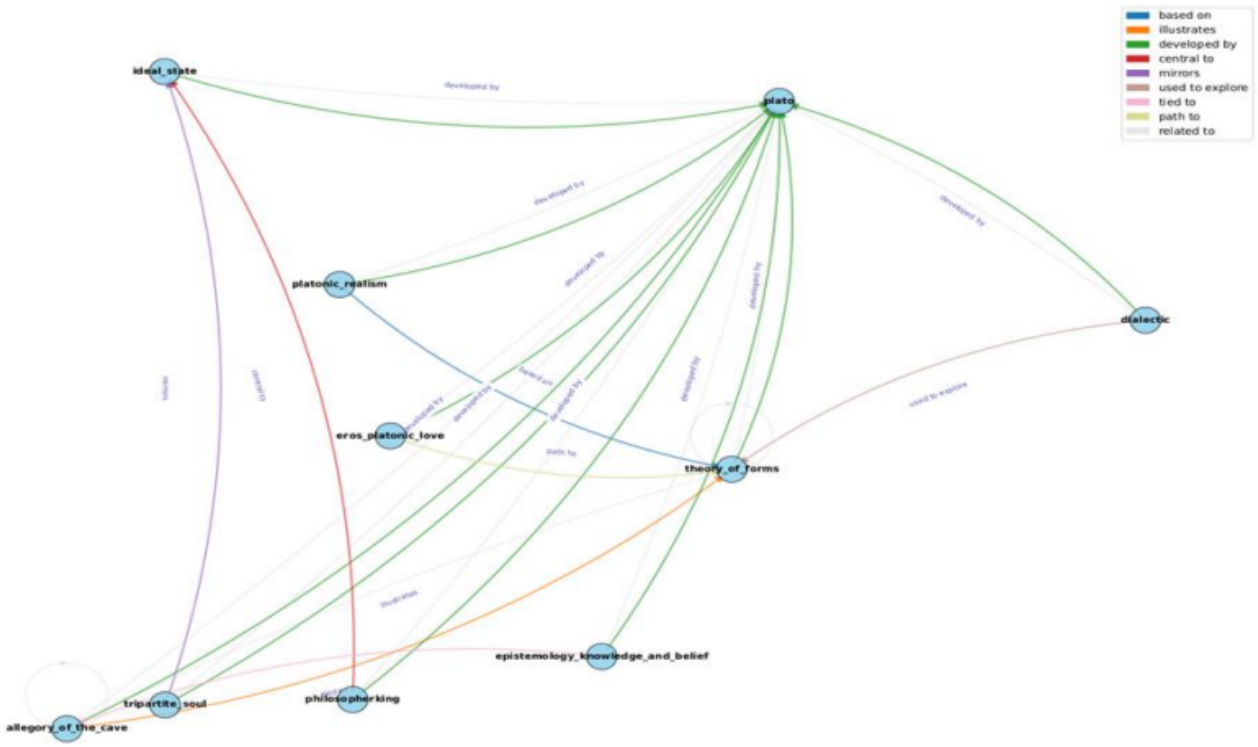

图 6: 从 Grok 提取的关于柏拉图的知识图谱。

[6] Buchanan, B. G., and Smith, R. G. Fundamentals of expert systems. *Annual review of computer science 3*, 1 (1988), 23–58.

[7] Chelli, M., Descamps, J., Lavoué, V., Trojani, C., Azar, M., Deckert, M., Raynier, J.-L., Clowez, G., Boileau, P., Ruetsch-Chelli, C., et al. Hallucination rates and reference accuracy of chatgpt and bard for systematic reviews: comparative analysis. *Journal of medical Internet research 26*, 1 (2024), e53164.

[8] Copeland, B. J. DENDRAL | expert system. *Encyclopædia Britannica Online*, 2008. https://www.britannica.com/technology/DENDRAL.

[9] Dam, S. K., Hong, C. S., Qiao, Y., and Zhang, C. A complete survey on llm-based ai chatbots. *arXiv preprint arXiv:2406.16937* (2024).

[10] Feigenbaum, E. A., and Buchanan, B. G. Dendral and meta-dendral: Roots of knowledge systems and expert system applications.

[11] Giarratano, J., and Riley, G. *Expert Systems: Principles and Programming*, 4th ed. Cengage Learning, 2005.

[12] Gou, J., Yu, B., Maybank, S. J., and Tao, D. Knowledge distillation: A survey. *International journal of computer vision 129*, 6 (2021), 1789–1819.

[13] Hart, A. Knowledge acquisition for expert systems. In *Knowledge, skill and artificial intelligence*. Springer, 1988, pp. 103–111.

[14] Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems 43*, 2 (2025), 1–55.

[15] Ji, Z., Yu, T., Xu, Y., Lee, N., Ishii, E., and Fung, P. Towards mitigating llm hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023* (2023), pp. 1827–1843.

[16] Kang, S., Lee, D., Kweon, W., and Yu, H. Personalized knowledge distillation for recommender system. *Knowledge-Based Systems 239* (2022), 107958.

[17] Kujanpää, K., Valpola, H., and Ilin, A. Knowledge injection via prompt distillation. *arXiv preprint arXiv:2412.14964* (2024).

[18] Leiser, F., Eckhardt, S., Knaeble, M., Maedche, A., Schwabe, G., and Sunyaev, A. From chatgpt to factgpt: A participatory design study to mitigate the effects of large language model hallucinations on users. In *Proceedings of Mensch und Computer 2023*. 2023, pp. 81–90.

[19] Li, L., Zhang, Y., and Chen, L. Prompt distillation for efficient llm-based recommendation. In *Proceedings of the 32nd ACM international conference on information and knowledge management* (2023), pp. 1348–1357.

[20] Liu, Y., Chen, K., Liu, C., Qin, Z., Luo, Z., and Wang, J. Structured knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 2604–2613.

[21] McDermott, J. R1 (xcon) at digital equipment corporation. *Artificial Intelligence Magazine 3*, 4 (1982), 40–51.

[22] Ni, Y., Zhu, H., Cai, P., Zhang, L., Qui, Z., and Cao, F. Cliniqa: highly reliable clinical question answering system. In *Quality of Life through Quality of Information*. IOS Press, 2012, pp. 215–219.

[23] Ogu, E. C., and Adekunle, Y. Basic concepts of expert system shells and an efficient model for knowledge acquisition. *International Journal of Science and Research 2*, 4 (2013), 554–559.

[24] Perković, G., Drobnjak, A., and Botički, I. Hallucinations in llms: Understanding and addressing challenges. In *2024 47th MIPRO ICT and electronics convention (MIPRO)* (2024), IEEE, pp. 2084–2088.

[25] Rawte, V., Sheth, A., and Das, A. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922* (2023).

[26] Renze, M. The effect of sampling temperature on problem solving in large language models. In *Findings of the association for computational linguistics: EMNLP 2024* (2024), pp. 7346–7356.

[27] Sriramanan, G., Bharti, S., Sadasivan, V. S., Saha, S., Kattakinda, P., and Feizi, S. Llm-check: Investigating detection of hallucinations in large language models. *Advances in Neural Information Processing Systems 37* (2024), 34188–34216.

[28] Stefik, M. *Introduction to knowledge systems*. Elsevier, 2014.

[29] Van Remoortere, P. Computer-based medical consultations: Mycin: Eh shortliffe: Published by north-holland, amsterdam and ny, 1976, 264 pages, us $19.95, isbn 0-444-00179-4, 1979.

[30] Ye, H.-J., Lu, S., and Zhan, D.-C. Generalized knowledge distillation via relationship matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 45*, 2 (2022), 1817–1834.