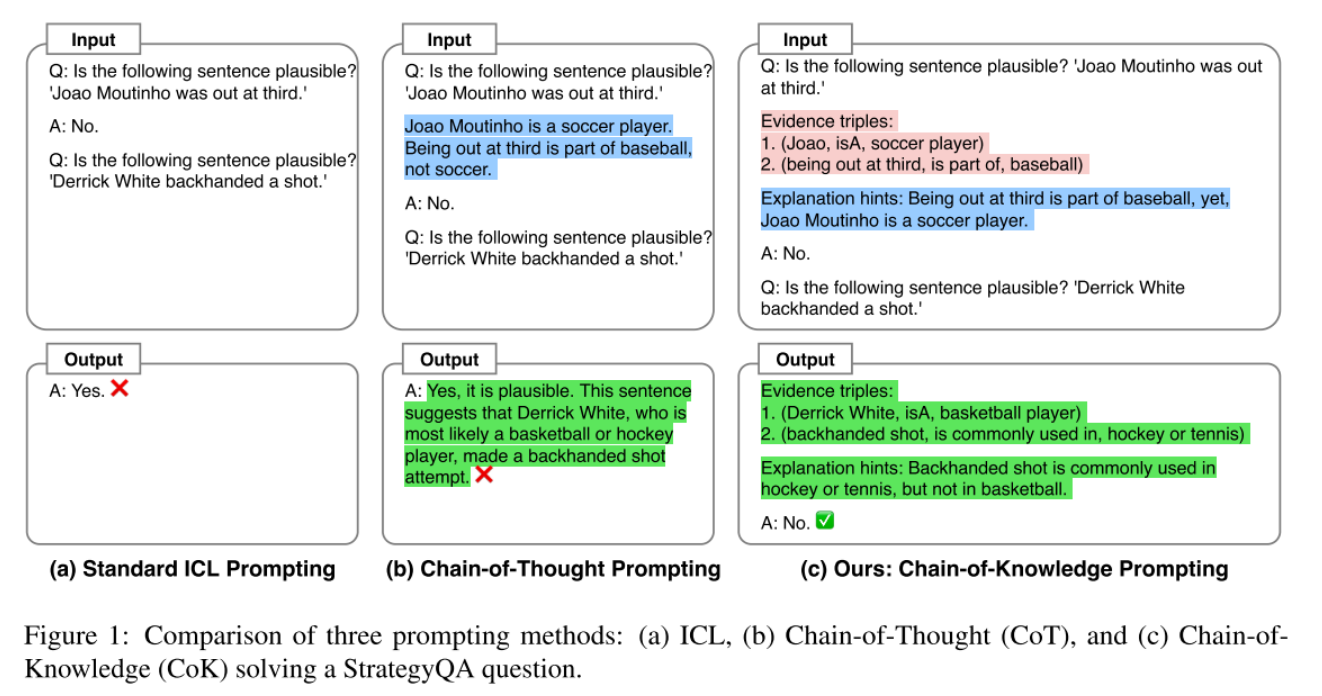
**Boosting Language Models Reasoning with Chain-of-Knowledge Prompting**

使用链式知识提示来增强语言模型的推理能力

最近，思维链(CoT)提示在复杂的推理任务中取得了成功，其目的是设计一个简单的提示，如“让我们一步一步地思考”或多个具有精心设计的基本原理的上下文示例，以引出大型语言模型(大型语言模型)来生成中间推理步骤。然而，产生的基本原理往往伴随着幻觉，形成不真实和不忠实的推理链。为了减轻这种脆弱性，我们提出了一种新的知识链(CoK)提示，我们的目标是激发大型语言模型以结构三重形式生成明确的知识证据。这是受到我们人类行为的启发，即在回答一个复杂的问题之前，我们可以在大脑中绘制思维导图或知识图作为推理证据。受益于CoK，我们还引入了一种F2-Verification方法来估计推理链在真实性和可信度方面的可靠性。对于不可靠的回应，可以指出错误的证据，促使大型语言模型反思。大量的实验表明，我们的方法可以进一步提高常识、事实、符号和算术推理任务的性能。

1.Introduction



大型语言模型（LLMs）已成功推动了许多自然语言处理（NLP）任务的最新技术发展，得益于超大规模训练语料库和计算资源。为了释放LLMs在未知任务上的适应能力，无需任何参数更新，基于上下文的学习（ICL）已成为一个蓬勃发展的研究课题，旨在通过在少量标记示例的条件下生成预测（图1（a））

最近的一系列研究表明，大型语言模型可以自发地将复杂的多步骤问题分解为中间推理链，通过“让我们一步一步地思考”这样的简单提示或精心设计的演示以及人工注释的基本原理来引出，这被称为思维链(CoT)提示(图1 (b)) 。这一发现很有趣，并且引起了轰动，因为CoT可能主要指定了一种输出空间/格式，该输出空间/格式规范了模型生成，以便在有序且与查询相关的情况下逐步查看。

尽管当前的大型语言模型（LLMs）表现出色，但它们容易产生幻觉（Ji等，2023年；Zhang等，2023年a），并提供不准确或不忠实的推理链，这不可避免地导致错误的结论。以图1为例。给定一个来自StrategyQA的查询：“以下句子是否合理‘Derrick White反手一击’”，标准的ICL和CoT给出了错误答案。一个推理步骤“Derrick White很可能是一名曲棍球运动员”是虚假的（事实上，Derrick White是一名篮球运动员），这导致了对问题的不准确推断。此外，当LLM生成逻辑上合理的推理链路，同时提供错误答案时，响应可能是不忠实的。

为了解决这些问题，我们提出了一种新颖的知识链（CoK）提示方法，通过一系列将显式结构知识证据与文本解释相结合的示例来增强LLM的推理能力。具体而言，CoK提示由两个组成部分组成（图1（c）），即证据三元组（CoK-ET）和解释提示（CoK-EH），其中CoK-ET是一个可以反映从查询到答案的整体推理证据的结构三元组列表，而CoK-EH是这些证据的解释。通过CoK提示构建具有上下文示例，我们首先对K个标记示例进行抽样，每个示例都可以与简单提示“让我们一步一步地思考”连接，以促使LLM生成推理链。然后，我们从外部知识库（KB）中检索一些结构三元组，并精心手动注释证据三元组，以获得设计良好的CoK提示。与标准的ICL和CoT类似，CoK提示可以被视为规范输出空间/格式的规则，并促使LLM生成明确的证据，而不仅仅是尝试生成模糊的文本推理链。此外，我们还提出了一种F2-验证策略来估计推理链的可靠性，以及关于事实性和忠实性的信任度，其中事实性是推理证据与基本事实知识之间匹配程度的量化，而忠实性是推理证据与最终答案的文本解释之间的一致程度。特别是对于不可靠的响应，可以指出错误的证据片段，促使LLM重新思考问题。我们设计了一种重新思考算法来实现这一目标。

我们对各种推理任务（例如常识、事实、算术和符号性推理）进行了实证评估，结果显示，结合F2-验证的CoK提示可以显著优于标准ICL和CoT提示。我们还将CoK提示与一些流行的策略（如自一致性）整合在一起。结果表明，这样的CoK可以作为一个即插即用的模块，进一步提高推理能力。

2.Related Work

针对LLMs进行上下文学习的提示。上下文学习（ICL）是因果语言建模的任务，允许LLMs通过一个精心设计的基于文本的提示进行零/少样本学习。ICL可以绕过模型参数更新，并通过对少量标记示例进行条件化而实现显著的性能。先前的研究探讨了ICL的一些影响方面。例如，输入-输出映射和模板格式，示例的不同选择和排列。为了提高ICL的有效性，一些新颖的方法已被提出，涉及元学习，提示和示例工程（。

思维链提示引发推理。最近，CoT提示被提出，利用推理和可解释信息来引导LLMs生成可靠且可解释的响应。一系列增强CoT的方法被提出，进一步提高推理能力。例如，Wang等人（2023d年）引入自一致性来通过边缘化采样的推理路径来抑制错误的基础问题，找到最一致的答案。Fu等人（2023年）和Besta等人（2023年）提出了逻辑思维图，让LLMs更好地推理。Lyu等人（2023年）将复杂问题转化为交错的自然语言或编程语言，使推理链忠实。Li等人（2023年）和Yao等人（2023a年）引入粗粒度或细粒度标签来验证推理链。与此不同的是，我们专注于减轻与推理链的事实性和忠实性相关的虚构问题。

3.Methodology

由CoT提示引发的推理链有时会出现错误，最终导致产生虚构的答案。我们将这一挑战归因于文本推理链：LLMs可能会强行生成符合CoT提示格式但在逻辑上模糊且得出错误答案的文本理由。为了解决这一挑战，我们提供了关于如何在两个方面增强LLM推理能力的具体解决方案：提示的推理引发格式和知识增强后验证。框架概述如图2所示。

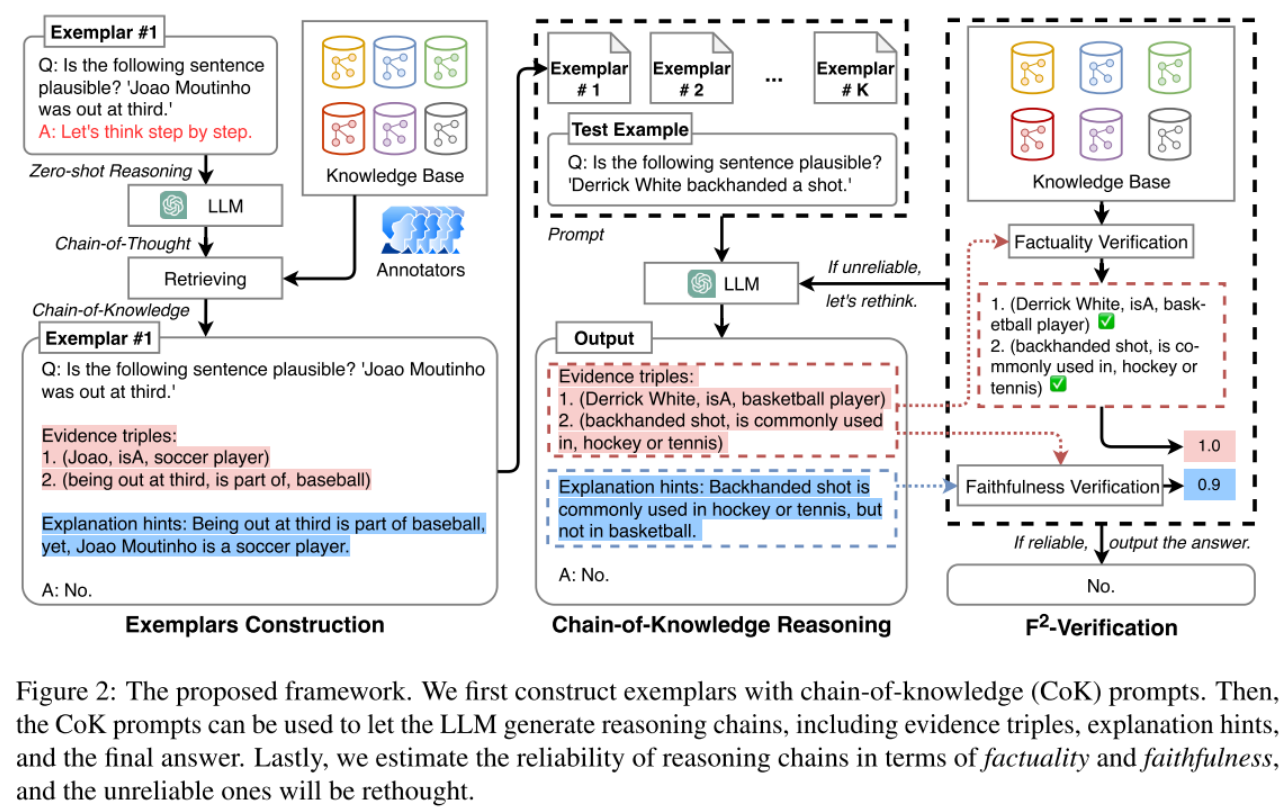


图2：提出的框架。首先，我们使用知识链（CoK）提示构建示例。然后，可以使用CoK提示让LLM生成推理链，包括证据三元组、解释提示和最终答案。最后，我们根据事实性和忠实性评估推理链的可靠性，不可靠的推理链将被重新考虑。

3.1 Chain-of-Knowledge Prompting

广泛认为，推理可以被建模为对现有知识系统的归纳和演绎。这受到人类行为的启发，人们绘制思维导图或知识地图来分析问题并找到通往答案的正确路径。幸运的是，我们可以采用知识库中的三元组的概念，可以被视为“（主体，关系，客体）”，以形式化推理链的显式证据。具体而言，我们提出了知识链（CoK）提示，以促进更好的提示LLMs的引发提示，它由两个主要组成部分组成，即证据三元组（CoK-ET）和解释提示（CoK-EH）。CoK-ET代表了多个三元组的列表，每个三元组代表了从LLMs探索出来支持逐步思考过程的知识证据。CoK-EH表示推理链的解释，类似于CoT。以图1为例，我们可以敦促LLM生成明确的证据片段来支持最终答案。

3.2 Exemplars Construction

在前期研究的启示基础上，ICL的表现取决于注释的基本原理。这表明，CoK提示的关键挑战在于构建带有结构证据三元组的文本基本原理。如图2所示，我们首先执行示例构建，以获得一个设计良好的任务特定提示。具体而言，我们随机选择K个问题作为基本示范。为了自动获取CoK-EH，我们通过零-shot CoT和一个简单的提示“让我们一步一步思考”为每个问题生成文本基本原理。另一个挑战是如何获得注释的CoK-ET，以更好地表达文本基本原理。为了找出答案，我们首先构建一个包括字典、常识、实体、事件、脚本和因果关系等六个领域的知识库K，这些领域以三元组的形式呈现。然后，我们直接使用检索工具来检索一些候选三元组。最后，我们邀请5名专业注释员根据检索到的三元组手动设计相应的CoK-ET。更多详细信息请参见附录C.1。

3.3 F2-Verification

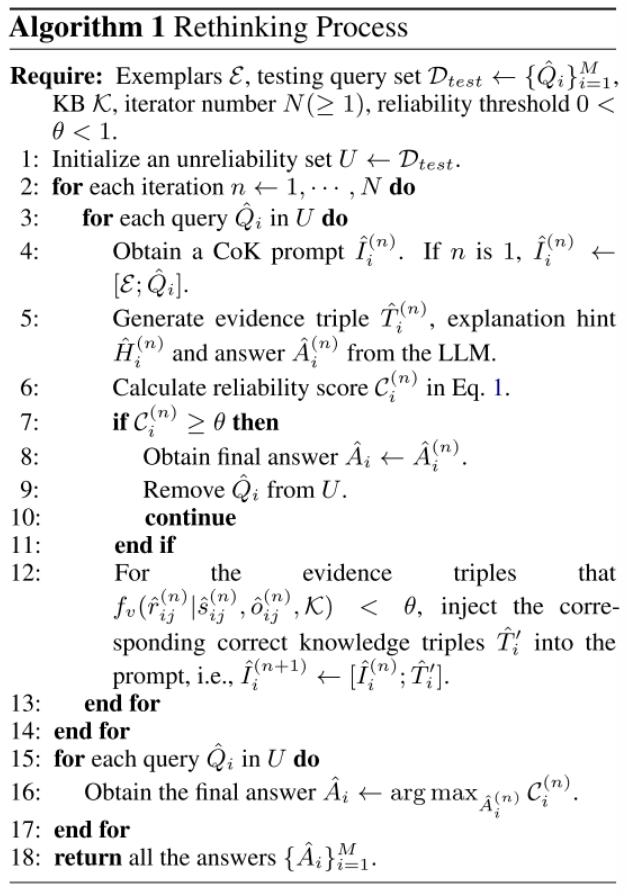
在示例构建之后，我们可以获得K个注释数据E = {(Qi, Ti, Hi, Ai)}Ki=1。符号上，Qi，Hi和Ai分别代表第i个示例的输入查询，解释提示和最终答案；每个都是令牌序列。Ti表示证据三元组的列表，其中包含多个知识三元组，即Ti = {(sij, rij, oij)}j，其中sij，rij和oij分别代表主体，关系和客体。给定一个测试查询输入^Qi，我们可以直接选择E的一个排列，并将它们与这个测试查询连接成一个线性序列^Ii = [E; ^Qi]，以促使LLM生成预测，即yˆik = arg maxσ(ˆyi(≤k)) P(y|ˆyi(<k), E, ^Qi)，其中P(y|·)是预测分布，ˆyik是第k个令牌，σ(·)表示解码策略（例如，温度抽样、束搜索和核采样），[·; ·]是连接操作。

由于示范的良好设计格式，从LLM导出的最终预测yˆi 包括一个证据三元组Tˆi 列表，一个解释提示序列ˆHi 和最终答案Aˆi。然而，LLM可能会生成虚构的基本原理，因此无法保证最终答案。我们将这个问题归因于两个因素：1）基本原理中的一些步骤可能与事实不符，导致错误，2）最终答案与推理链之间的关系仍然模糊，使回答不够可信。为了缓解这些缺点，我们提出了F2-Verification 来估计答案的可靠性，以实现对事实性和忠实性的双重验证。

**事实性验证**。我们首先验证事实性，这可以被视为每个生成的证据三元组与来自知识库的基准知识之间的匹配程度。具体而言，我们首先定义一个函数fv(ˆrij|ˆsij, ˆoij, K)来表示每个证据的事实性。我们设计了两种不同的fv策略。1）精确验证。我们可以基于主体ˆsij和客体ˆoij检索所有相关的三元组，然后找到生成的关系ˆrij是否存在，即fv(ˆrij|ˆsij, ˆoij, K) = I((ˆsij, ˆrij, ˆoij) ∈ K)。2）隐式验证。对于在知识库中不存在的三元组，可以进行纠正。因此，我们可以将事实性验证转化为一个图完成任务，预测三元组是否为真。为简单起见，我们使用TransR（Lin等人，2015）对知识库K进行预训练，并使用现成的能量函数为每个证据三元组分配分数，即fv(ˆrij|ˆsij, ˆoij, K) = ||s(r,c)ij + rc − o(r,c)ij||2^2 + α||rc − rij||2^2，其中α > 0是平衡因子，|| · ||2^2是Frobenius范数。s(r,c)ij = sijMr和o(r,c)ij = oijMr分别表示主体sij和客体oij在关系空间r中的投影表示。sij、rij和oij ∈ Rd分别是ˆsij、ˆrij和ˆoij的d维嵌入。rc ∈ Rd是关系r的原型嵌入。Mr ∈ Rd×d是关系r的可训练投影矩阵。我们在我们的框架中结合了这两种策略。如果证据三元组存在于K中，我们将使用精确验证策略分配一个得分，否则我们将使用隐式验证策略。

忠实性验证。根据Jacovi和Goldberg（2020）以及Lyu等人（2023）的定义，如果从模型中得出的推理过程可以准确地用解释来表达，我们称之为忠实。基于思维链提示的先前工作在验证忠实性方面存在困难，因为缺乏足够的证据来理解解释与答案之间的关系（Ye和Durrett，2022）。因此，我们提出了一种忠实性验证方法来找出这些情况。具体而言，给定一个测试查询Qˆi，一个证据三元组列表Tˆi和最终答案Aˆi，我们直接将它们连接成一个新的序列Hˆ′i。我们利用预先构建的句子编码器SimCSE来计算Hˆ′i和Hˆi之间的相似性。我们将这个函数表示为fu(ˆHi|ˆH′i = [ˆQi; ˆTi; ˆAi]) = SimCSE(ˆHi, ˆH′i)。最后，对于每个查询ˆQi，我们可以获得一个代表理性对答案可靠性的分数Ci（0 < Ci < 1）：Ci = γ \* 1/|ˆTi| \* Σfv(ˆsij|ˆrij, ˆoij, K) + (1 − γ) \* fu(ˆHi|ˆH′i = [ˆQi; ˆTi; ˆAi])，其中0 < γ < 1是平衡因子，默认设置为0.5，|ˆTi|是三元组的数量。

3.4 Rethinking Process



F2-验证有助于确保模型生成的三元组和解释的事实性和忠实性。在验证范围之外，我们可以通过采用重新思考过程进一步提高LLMs的性能，如算法1所示。我们初始化一个可靠性阈值θ（0 < θ < 1）、迭代次数N和一个不可靠集合U。测试集Dtest中的所有查询最初被视为不可靠。对于第n次迭代中U中的每个查询ˆQi，我们通过结合演示E和查询ˆQi获得CoK提示ˆI(n)i。该提示可用于引导LLM生成一组证据三元组ˆT(n)i、解释提示Hˆ(n)i和最终答案Aˆ(n)i。所提出的重新思考算法将允许LLMs通过计算等式1中的得分评估理性（即Tˆ(n)i和Hˆ(n)i）的可靠性。如果C(n)i未低于阈值θ，则不再将条目视为不可靠，随后导致最终答案Aˆi。相反，假设C(n)i未达到θ。在这种情况下，我们可以选择得分较低的证据三元组，并在下一次迭代中从知识库中注入相应的正确知识三元组到CoK提示ˆI(n+1)i（第12行）。这种动态生成评估过程将持续进行，直到U中的所有条目被视为可靠，或达到最大迭代次数N。对于达到最大迭代次数但没有任何三元组的可靠性得分超过θ的情况，将选择可靠性得分最高的三元组进行推断（第15-17行）。

4 Experiments