文章编号：1003-0077(2026)01-0030-10

面向小规模大语言模型推理优化的推理路径排序方法

李 俊[1]，白 宇[1,2]，刘雨婷[1]

(1.沈阳航空航天大学 计算机学院，辽宁 沈阳 110136；
2.多语言协同翻译技术国家地方联合工程实验室，辽宁 沈阳 110136)

摘 要：尽管大语言模型(LLM)在自然语言处理领域取得巨大成功，但是伴随其千亿级参数规模的训练也产生了巨大的计算成本。小规模大语言模型(SLLM)作为低资源场景下实现 LLM 部署的可替代方案，但任务处理能力与 LLM 尚存在明显差距。尽管上下文学习(ICL)等提示方法在一定程度上提升了 SLLM 的问题处理能力，但基于人工构建的提示往往需要参与者具备特定的专业领域知识，这给 LLM 的普适推广带来了挑战。针对以上问题，该文提出了一个基于 SLLM 的问题推理框架，通过在推理路径生成和答案生成两个阶段之间引入基于逐步语义验证器(SSVRP)的推理路径排序选择机制，在无人干预情况下实现 SLLM 推理能力提升。实验结果表明，SSVRP 有效地增强了 SLLM 的推理性能，在 4 个推理任务中的平均准确率分别达到了 54.3%，90.6%，64.3%和 63.7%，并在其中的 3 个推理任务中都取得了最新的 SOTA 结果。

关键词：逐步语义验证器；小规模大语言模型；上下文学习；推理路径排序

中图分类号：TP391 **文献标识码**：A

A Reasoning Path Ranking Method for Reasoning Optimization of Small-scale Large Language Models

LI Jun[1], BAI Yu[1,2], LIU Yuting[1]

(1.School of Computer Science, Shenyang Aerospace University, Shenyang, Liaoning 110136, China;
2.National & Local Joint Engineering Laboratory of Multi-Language Collaborative Translation Technology, Shenyang, Liaoning 110136, China)

Abstract: As an alternative to LLM deployment in low-resource scenarios, Small-scale Large Language Models (SLLM) still have a significant gap in task processing capabilities compared to LLM. This paper proposes a question reasoning framework for SLLM. By introducing a reasoning path ranking selection mechanism based on Step-Semantic Verifier on Reasoning Paths (SSVRP) between the two stages of reasoning path generation and answering generation, the reasoning capability of SLLM is improved without human intervention. Experimental results show that SSVRP effectively enhances the reasoning performance of SLLM, with average accuracy of 54.3%, 90.6%, 64.3% and 63.7% in four reasoning tasks, respectively, which are the latest SOTA results in three of the reasoning tasks.

Keywords: step-semantic verifier on reasoning paths; small-scale large language models; in-context learning; reasoning path ranking

0 引言

大型语言模型(Large Language Models, LLM)，如 OpenAI 的 ChatGPT[1]、谷歌的 Gemini[2] 以及 Meta 的 LLaMA[3]等，在推理问答[4]、语言翻译[5]、情感分析[6]和文本生成[7]等自然语言处理相关的多种任务中取得了巨大的成功，对人工智能领域的发展产生了深远的影响。

LLM 通常具有复杂的模型结构以及千亿级的

收稿日期：2024-08-20 **定稿日期**：2024-09-24
基金项目：国家自然科学基金(U1908216)

参数规模，这使得在本地的生产环境中部署运行 LLM 需要大量的计算资源成本[8]。以 4-Bit 量化的 LLaMA 模型为例，高效执行 LLaMA-7B 模型至少需要配备 6 GB VRAM 的 GPU，而要运行 LLaMA-65B 或 70B 模型，则至少需要 40 GB VRAM 的 GPU[9]。表 1 详细列出了不同参数规模的 LLaMA 模型对 GPU 资源的需求和最低部署成本。

表 1　4-Bit 量化的 LLaMA 模型对 GPU 的需求和最低部署成本

LLaMA 模型系列	最小 VRAM 要求/GB	价格/美元
LLaMA/Llama-2 7B	6	280
LLaMA/Llama-2 13B	10	3 00
LLaMA/Llama-2 33B	20	2 800
LLaMA/Llama-2 65B/70B	40	5 600

近期的工作[10-11]表明，小规模大语言模型(Samll-scale Large Language Model，SLLM)相对于更复杂的模型版本具有若干优势。与 LLM 相比，SLLM 通常具有更少的参数、更小的尺寸、更快的训练和推理速度，需要更少的内存和外存，并且消耗更少的能量。这些优势可以直接为在本地部署此类模型的组织节省成本。然而，值得注意的是，SLLM 的参数量减少可能导致其总体上与 LLM 相比，问题处理能力相对较低。例如，大多数 SLLM，即使经过微调，也难以在处理文摘任务方面超越零样本 LLM[12]。

上下文学习(In-Context Learning，ICL)技术的核心在于利用一系列的输入输出对作为示例，通过给 LLM 提供有限的示例来指导 LLM 完成特定任务。该技术使得模型能够在无须参数调整的情况下预测准确地输出。尽管 ICL 提示同样成功地增强了 SLLM 在推理任务中的能力，但提示的构建通常需要人工参与，并需要参与者具备特定的专业领域知识，这给 SLLM 的普适推广带来了挑战。

在应对复杂的数学或逻辑推理问题时，将一个问题拆分成一系列的中间步骤，通常能帮助 LLM 找到最终答案。Wei 等人[13]提出思维链(Chain-of-Thought，CoT)，并通过向 LLM 提供一些解题的示例，证实了 CoT 提示能够促使 LLM 将复杂的解题过程拆分为一系列的解题步骤，从而逐步引导 LLM 生成问题的正确答案。CoT 提示作为 ICL 提示中的一种主流提示方法，在推理任务中表现出良好的效果。

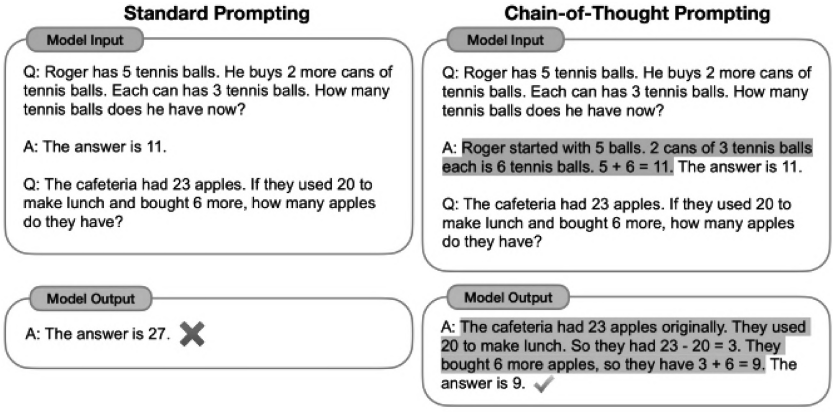

图 1　标准的输入-输出提示和 CoT 提示[13]

在实际应用中，与使用“Let's think step by step.”提示的 Zero-Shot-CoT[14]方法相比，通过人工手动构建的 Manual-CoT[13]显示出更优越的性能。然而，这种高性能依赖于精心制作的示例，而创建这些示例依赖人工。Zhang 等人[15]提出 Auto-CoT 范式，通过自动化方式构建包含问题及其推理链的示例，旨在克服手动构建示例的限制。Wang 等人[16]提出了 Self-Consistency，通过在多个推理路径中投票选择最一致的答案来提高解答的准确性，这种方法在多项算术和常识推理任务中取得了显著效果。然而，这种投票机制在面对极其多样化的推理路径时可能无法奏效。为了令 LLM 进行有效的推理，相关研究对不同的推理路径生成和选择进行了探索。Li 等人[17]认为，错误推理路径中的每一个步骤并非全无价值，因此提出利用 DIVERSE 验证器对 LLM 生成的各种推理路径上的每一步进行评估，以挑选出有助于 LLM 推理的最佳推理路径。

本文探索如何在无须人工构建任何提示的情况下，以最小的成本增强 SLLM 来模拟 LLM 在推理任务中的性能。本文的主要贡献如下：

(1) 提出了一个新的框架,该框架与 CoT 和 Auto-CoT 相比,在两个 SLLM 之间引入基于逐步语义验证器(Step-Semantic Verifier on Reasoning Paths, SSVRP)的排序模型(图 2),SSVRP 实现了推理路径的选择优化,从而增强 SLLM 在推理任务中的性能。

(2) 提出了一种基于对比学习的逐步语义验证器训练方法,所有训练数据均由 SLLM 生成。在 4 个公开的推理任务数据集上进行的实验表明,答案生成的平均准确率分别达到了 54.3%,90.6%,64.3%和 63.7%,并在其中 3 个推理任务中都取得了最新的 SOTA 结果,达到了与基于 LLM 的 Auto-CoT 方法相当的性能。

(3) 讨论了 SLLM 在推理路径生成和答案生成阶段的性能。我们发现不同的 SLLM 在推理路径生成阶段存在显著差异,而在答案生成阶段,当提供相同的推理路径时,所有模型的表现几乎相同。这表明推理路径的质量与答案的质量直接相关,表明对推理路径进行优化能够带来更准确的答案。

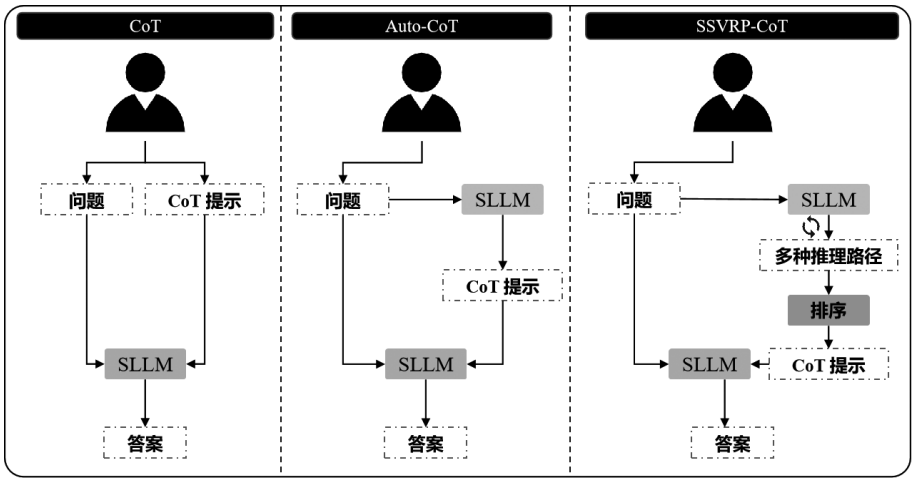

图 2 本文方法与其他 Auto-CoT 方法

1 相关工作

Wei 等人[13]提出了一种 CoT 提示方法,以解决 LLM 在理解复杂问题方面的局限性。他们手动构建 CoT 提示示例,逐步引导 LLM 将复杂问题分解为子问题,从而显著提高 LLM 的表现。Fu 等人[18]提出了一种根据所需推理的复杂性来选择提示的创新方法。他们优先考虑需要更复杂推理步骤的提示,从而促进详细的 CoT 流程。为了增强 CoT 提示,一些研究将输入-输出关系分解为增量推理阶段。他们采用了上下文学习(ICL),让 LLM 在不同的推理阶段接收不同的提示。Self-Ask[19]技术允许 LLM 生成与初始输入相关的补充查询,允许 LLM 自行查询和响应,将这些查询和响应集成到 CoT 提示中。iCAP[20]引入了一个动态的上下文感知提示系统,可以调整每个推理阶段的上下文。此外,Least-to-Most[21]提示方法涉及两个阶段的 ICL 过程,首先将复杂问题简化为子问题,然后按顺序解决这些子问题,同时将前期阶段的答案融入正在进行的上下文中。

Chung 等人[22]观察到,使用包含中间推理步骤的数据训练 SLLM 可以显著提高 SLLM 的性能。STaR[23]可以使其通过自我生成的原因进行自我改进,从而提高 SLLM 在推理任务中的性能。SpecialFT[18]使用 LLM 作为教师模型,并利用知识蒸馏将推理能力从 LLM 转移到 SLLM。Orca[24]学习模仿 LLM 的推理过程,包括解释性痕迹、逐步的思维过程和其他复杂的指令。这些方法的共同特点是调整 SLLM 的参数并向 LLM 学习。

2 逐步语义验证器

借鉴 LLM 的语言理解和生成能力,我们使用了两个 SLLM。如图 3 所示,其中 SLLM.I 的主要功能是生成问题的推理路径,将问题及一个初始提示"Let's think step by step."作为 SLLM.I 的输入,通过多次迭代使 SLLM.I 生成多条不同的候选推理路径。将利用语义验证器选出的最佳推理路径作为 SLLM.II 的输入,生成问题的最终答案。

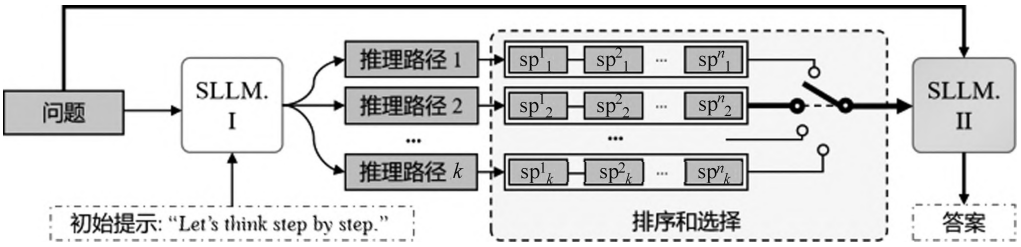

图 3　基于推理路径排序的问题推理

算法 1 概述了从一组候选推理路径中选择最优推理路径的过程。

2.1　推理路径的排序和选择

对推理路径进行排序和选择是构建 CoT 提示的一个关键步骤，这一过程对于 SLLM 在逻辑推理和问题解决中至关重要。对推理路径进行排序和选择可以有效防止 SLLM 探索低质量或不相关的推理路径，从而提高 SLLM 在推理过程中的效率。

对于给定的问题 q，使用“.”将其拆分为多条信息，记作 $\mathrm{q}=[\mathrm{q}_1,\cdots,\mathrm{q}_p,\cdots,\mathrm{q}_n]$，其中，$q_p$ 表示问题 q 中的第 p 条信息。假设 SLLM.I 为问题 q 生成 k 条推理路径，使用“.”将每一条推理路径其拆分为多个推理步骤，构成序列 $R=[\mathrm{sp}_1^{(1)},\cdots,\mathrm{sp}_{m_1}^{(1)},\cdots,\mathrm{sp}_j^{(i)},\cdots,\mathrm{sp}_1^{(k)},\cdots,\mathrm{sp}_{m_k}^{(k)}]$，$\mathrm{sp}_j^{(i)}$ 表示第 i 条推理路径中的第 j 个推理步骤，$i=1,\cdots,k$，$j=1,\cdots,m_i$。在这里，最优的推理路径是指能够引导 SLLM.II 生成正确答案的推理路径。考虑到在实际场景中，SLLM.II 能否利用推理路径得出正确答案具有不确定性，我们把与问题提供的语义信息最接近的推理路径作为最佳推理路径，其在序列 R 中的位置表示为 Pos^*。

$$\mathrm{Pos}^*=\underset{k}{\operatorname{argmax}}\Big[\sum_{p=1}^{n}\sum_{j=1}^{m_k}\mathrm{sim}(q_p,\mathrm{sp}_j^{(k)})/m_k\Big] \tag{1}$$

其中，函数 sim(·)的作用是预测问题 q 中第 p 条信息和第 k 条候选推理路径中第 j 个推理步骤之间语义相似性。逐步语义验证器的训练和用于最佳推理路径排序和选择的过程如图 4 所示。

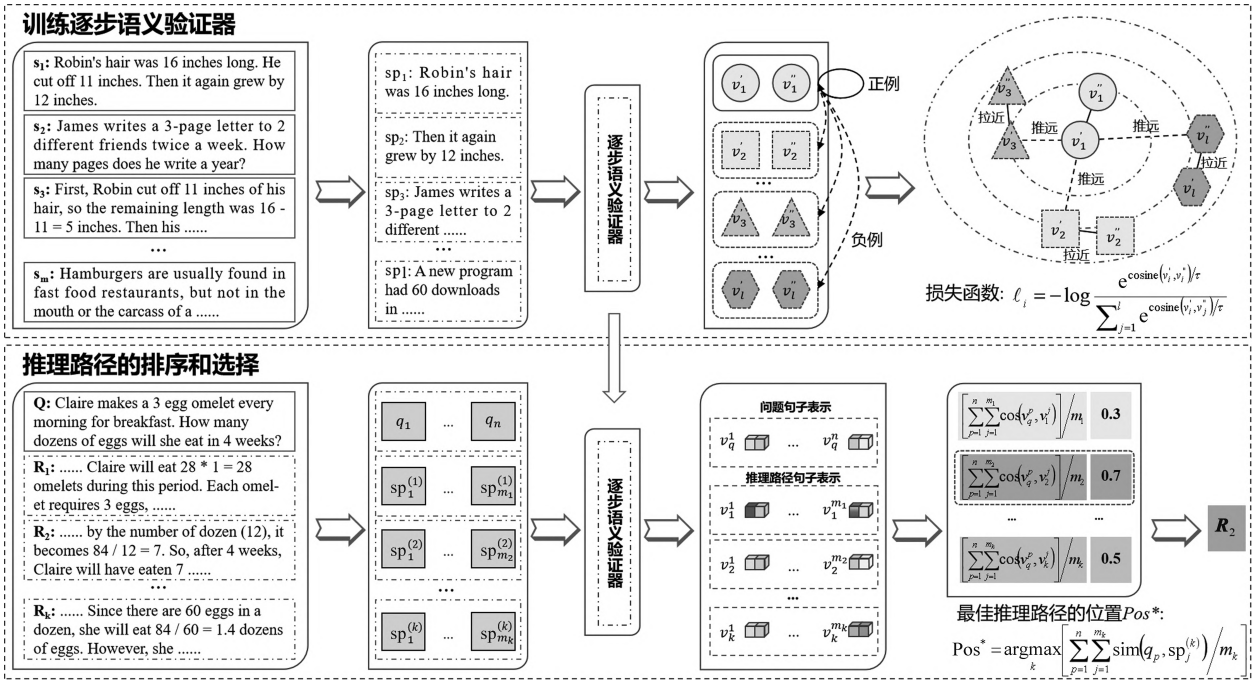

图 4　逐步语义验证器的训练和用于最佳推理路径排序和选择的过程

2.2　逐步语义验证器

为了实现函数 sim(·)，我们提出了一个逐步语义验证器(SSVRP)，SSVRP 将问题中的每一条信息和候选推理路径中的每一个推理步骤进行编码，使其映射到潜在语义空间。令 $\boldsymbol{V}_q^p$ 表示问题 q 的

第 p 个信息编码后的表示向量，$\boldsymbol{V}_i^j$ 表示第 i 条推理路径的第 j 个推理步骤编码后的表示向量，通过计算 $\boldsymbol{V}_q^p$ 和 $\boldsymbol{V}_i^j$ 之间夹角的余弦值实现函数 sim(·)的输出。

为了学习推理路径中每一个推理步骤的语义特征，在不使用 LLM 和人工干预的情况下，利用 SLLM 创建用于训练逐步语义验证器的数据集。为了保证推理路径的多样性，我们采用迭代的方式利用多个不同的 SLLM 为训练问题集中的每一个问题生成多条推理路径。在这个推理路径的生成过程中，使用了初始化提示“[Question] + Let's think step by step.”。

逐步语义验证器的训练遵循 SimCSE[25] 中无监督的对比学习框架，将问题本身视为一条推理路径并入训练数据集。我们将该训练数据集表示为句子集合 $\{\mathrm{sp}_j\}_{j=1}^{l}$，采用 dropout 噪声和 $\mathrm{sp}_j^+ = \mathrm{sp}_j$ 作为数据增强的一种形式。将同一个句子两次经过 dropout 操作，得到句子的两个不同表示向量互为正例，不同句子的嵌入表示互为负例。

$$v_j^z = f_\theta(\mathrm{sp}_j, z) \tag{2}$$

其中，z 是 dropout 的随机掩码(默认 $p=0.1$)，我们只需将相同的输入分两次提供给编码器并获得两个不同的嵌入 v'，v''。在这项工作中，我们使用的对比学习损失函数与 SimCSE 相同，如式(3)所示。

$$l_i = -\log \frac{\mathrm{e}^{\mathrm{cosine}(v_i', v_i'')/\tau}}{\sum_{j=1}^{l} e^{\mathrm{cosine}(v_i', v_j'')/\tau}} \tag{3}$$

其中，τ 是温度超参数，$\mathrm{cosine}(v', v'')$ 是余弦相似度 $\frac{v'^{T} v''}{\|v'\| \|v''\|}$。我们使用目标函数 l_i 对预训练语言模型 DeBERTa-base[26] 进行微调，并利用微调后的模型对输入的句子进行编码，得到问题中的每一条信息和候选推理路径中的每一个推理步骤在潜在语义空间中的向量表示。

3 实验

3.1 实验步骤

3.1.1 数据集描述

本文实验采用 4 个公开数据集测试 SLLM 在算术推理和常识推理任务上的表现。其中，GSM8K[27]、MultiArith[28] 以及 SVAMP[29] 主要用来评估 SLLM 解决复杂的多步算术问题的能力，CommonsenseQA[30] 数据集用来评估 SLLM 解决常识推理问题的能力。实验数据集的详细信息如表 2 所示。

表 2 实验数据的详细信息

数据集	问题类型	训练数据的规模	测试数据的规模
GSM8K	数学应用题	7 473	1 319
MultiArith	数学应用题	420	180
SVAMP	数学应用题	700	300
CommonsenseQA	常识单选题	9 741	1 221

3.1.2 训练数据

本文使用的训练数据集包含 8 593 个算术推理问题和 9 741 个常识推理问题。为了确保生成的推理路径的多样性，我们利用 3 个不同的 SLLM(LLaMA2、Baichuan2 和 ChatGLM3)为每个问题生成 9 个推理路径，每个 SLLM 生成 3 个推理路径。由此，共生成 77 257 个算术推理问题的推理路径和 87 669 个常识推理问题的推理路径。采用分隔符“.”将推理路径句子拆分成多个推理步骤，并通过随机采样构造算术推理问题推理步骤集合和常识推理问题推理步骤集合。

3.1.3 对比模型

对比模型采用 4 个基于 LLM 的增强模型和 9 个基于 SLLM 的增强模型。基于 LLM 的增强模型包括分别使用 Greedy Decode 和 Self-Consistency 提示增强的 GPT-3(davinci)、分别使用 Greedy Decode 和 Self-Consistency 提示增强的 GPT-3 (text-davinci-002)。基于 SLLM 的增强模型包括分别使用 Greedy Decode、Self-Consistency和 DIVERSE 提示增强的 FlanT5-XL (3B)，使用 Greedy Decode、Self-Consistency 和 DIVERSE 提示增强的 LLaMA2(7B)，使用 Greedy Decode、Self-Consistency 和 DIVERSE 提示增强的 Baichuan2(7B)，使用 Greedy Decode、Self-Consistency 和 DIVERSE 提示增强的 ChatGLM3(6B)。这些比较模型的组件描述如下：

算法 1：推理路径的排序和选择

输入：A question：q，Number of Reason paths generated by SLLM-I：k.
输出：The Optimal Position in the List of Reasoning Paths Pos^*.

```
Initial the prompt:P ← "Let's think step by step."
Initial the Optimal position:Pos* ← 0
Initial the Optimal value:Score* ← −1
S[]← [Q, SLLMI(Q,P,k)]                    //SLLMI 生成问题的推理路径
q[] ← Split Q by "."
for each reasoning path R_i in S/Q do
    R_i[] ← Split R_i by "."
    Score_i ← 0
    for each sentence q_p in q do
        Score_{q_p} ← 0
        v_q^p ← Semantic_Verifier(q_p)
        //计算此条推理路径中所有推理步骤与其中一个问题信息的语义相似度的和
        for each sentence sp_j^(i) in R_i do
            n← 0
            v_i^j ← Semantic_Verifier(sp_j^(i))
            Score_{sp_j} ← cosine(v_q^p, v_i^j)
            Score_{q_p} ← Score_{q_p} + Score_{sp_j}
        end for
        Score_i ← Score_{q_i} + Score_i
    end for
    Score_i ← Score_i / n
    //找出最佳推理路径的位置
    if Score_i > Score* then
        Score* ← Score_i
        Pos* ← i
    end if
end for
return Pos*
```

（1）LLM

GPT-3[31]拥有 1 750 亿的参数，是 OpenAI 开发的开源大型语言模型。它以高精度和复杂性执行各种自然语言处理任务。本文引用了 Li 等人[17]报告的两个公共引擎 davinci 和 code-davinci-002 的实验结果。

（2）SLLM

FlanT5-XL(3B)[32]是 T5 模型的一种，XL 版本的 FlanT5 模型具有 30 亿参数量。

LLaMA2(7B)[3]是一个经过预训练和微调的大型语言模型，具有 70 亿参数。它是 Meta 开发的 LLaMA2 的最小版本。

Baichuan2(7B)[33]是一个包含 70 亿参数的大规模多语言的语言模型，是百川智能开发的 Baichuan2 的最小版本。

ChatGLM3(6B)[34]是一个轻量级开源预训练对话模型，包含 60 亿参数。它是 ChatGLM3 的最小版本，ChatGLM3 是中英文训练的双语大语言模型，属于智普 AI 开发的大型语言模型系列。

（3）提示增强方法

Greedy Decode[14]是一种将问题与初始提示“Let's think step by step.”一起输入 LLM 的方法。这种方法鼓励 LLM 逐步解决问题，从而得出最终答案。

Self-Consistency[16]使用 Few-Shot-CoT 提示生成多个推理路径和对应的答案，然后选择频率最高的答案作为最终答案。

DIVERSE[17]是最近研究中最先进的方法，可增强 LLM 的推理能力。此方法需要 LLM（code-davinci-002）生成的推理路径的数据集来训练验证器。

3.1.4　超参数设置

在 SLLM.I 生成候选的推理路径时，期望让 SLLM.I 生成的推理路径具有多样性，因此开启了 SLLM.I 的随机采样模式。在 SLLM.II 回答问题时，期望让 SLLM.II 生成更一致的内容，因此在此

阶段关闭了 SLLM.II 的随机采样模式。在训练阶段,我们引入了多头自注意力机制,设置 num_heads 的数量为 8,其他超参数设置如表 3 所示。

表 3 SLLM 和 DeBERTa 的超参数设置

模型	超参数	取值			
SLLM	do_sample	SLLM.I	True	SLLM.II	False
	top_k		10		1
	temperature		0.7		0
	k		3		—
DeBERTa	dropout	0.1			
	num_heads	8			
	τ	0.05			
	learning_rate	2×10^{-5}			
	batch_size	16			
	max_sequence_length	256			

3.2 实验结果

本文采用准确率(Accuracy)作为基于推理路径排序的问题推理框架在算术推理和常识推理任务中的性能评价指标,实验结果如表 4 所示。

3.3 实验分析

3.3.1 逐步语义验证器的有效性

为了验证基于 SSVRP 进行推理路径选择的有效性,将 SSVRP 与推理路径的随机选择方法进行比较。此外,为了验证将推理路径分割为多个推理步骤的必要性,将采用推理步骤进行语义验证的 SSVRP 与采用推理路径进行语义验证的 SVRP 方法进行比较。

SVRP 方法与 SSVRP 方法的主要区别在于,SVRP 方法不对推理路径按步骤进行拆分。具体来说,SLLM-I 模型会为每一道题生成多个推理路径,每一条推理路径都是一个包含多个解题步骤的解题过程。我们将题目信息和完整的推理路径作为训练语义验证器的训练数据。

如图 5 所示的实验结果表明,与随机选择推理路径的方法相比,在 ChatGLM3(6B)上,SSVRP 在多个推理任务中的性能均优于随机方法和 SVRP 方法,说明 SSVRP 的有效性。

表 4 SLLM 在不同的推理任务中的有效性比较。 (单位:%)

方法	GSM8K	MultiArith	SVAMP	CommonsenseQA	Average
GPT-3 (davinci, 175B):					
Greedy Decode	8.7	31.4	21.2	48.2	27.4
Self-Consistency	18.9	68.6	44.6	57.4	47.4
GPT-3 (text-davinci-002, 175B):					
Greedy Decode	37.1	70.7	60.0	65.5	58.3
Self-Consistency	58.2	88.4	78.2	72.9	74.4
Flan T5-XL(3B):					
Greedy Decode	5.4	10.0	8.7	77.8	25.5
Self-Consistency	2.9	6.1	7.3	**81.9**	24.6
DIVERSE (LLM)	**6.4**	14.4	11.3	79.3	27.9
SSVRP (SLLM)	6.2	**16.7**	**14.0**	80.6	**29.4**
LLaMA2 (7B):					
Greedy Decode	24.5	68.3	39.7	43.7	44.1
Self-Consistency	22.9	68.9	41.3	43.8	44.2
DIVERSE (LLM)	22.8	**69.7**	**44.3**	45.6	45.5
SSVRP (SLLM)	**25.9**	66.1	42.0	**48.7**	**45.7**
Baichuan2 (7B):					
Greedy Decode	30.9	74.4	48.7	55.4	52.4
Self-Consistency	29.0	79.4	47.3	57.4	53.3
DIVERSE (LLM)	**34.1**	80.6	**54.7**	58.5	56.9
SSVRP (SLLM)	33.5	**82.6**	53.7	**59.1**	**57.2**
ChatGLM3 (6B):					
Greedy Decode	52.9	86.7	59.3	60.8	64.9
Self-Consistency	45.8	93.9	60.0	61.4	65.3
DIVERSE (LLM)	52.7	**94.6**	62.3	62.7	68.1
SSVRP (SLLM)	**54.3**	90.6	**64.3**	**63.7**	**68.2**

注:最好结果用黑体表示,第二好的结果用下划线表示。

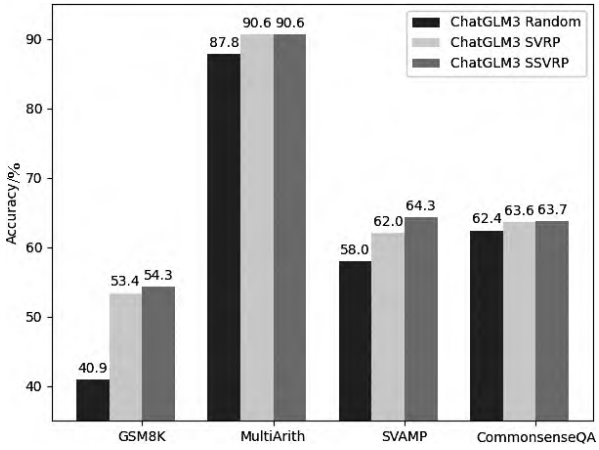

图 5　在 ChatGLM3 上的推理路径选择方法的比较

3.3.2　推理路径的增强效果

实验结果表明，与 Greedy Decode、Self-Consistency 和 DIVERSE 方法相比，在 LLaMA2(7B)上，SSVRP 在推理任务中的平均准确率分别提高了3.5%，3.3%和 0.4%，在 Baichuan2(7B)上，分别提高了 8.4%，6.8%和 0.5%，在 ChatGLM3(6B)上，分别提高了 4.8%，4.3%和 0.2%。此外，在更小参数量的 FlanT5-XL(3B)上，SSVRP 在推理任务中的平均准确率分别提高了 13.3%，16.3%和 5.1%。表明 SSVRP 方法对于 100 亿参数以下的 SLLM 是有效的。与不同提示增强方法(Greedy Decode 和 Self-Consistency)在 davinci 和 text-davinci-002 上的比较结果表明，SSVRP 增强的 SLLM 能够达到 LLM 的推理效果。

为了进一步研究 SLLM.I 生成的推理路径的质量对 SLLM.II 推理能力的影响，本文进行了 SLLM.I 和 SLLM.II 的组合实验，实验结果如表 5 所示。

实验结果表明，当提供相同的推理路径时，各个模型(SLLM.II)在答案生成阶段的表现几乎相同。这也表明，推理路径的优劣是影响问题推理结果的主要因素，有效的推理路径排序与选择方法有助于提升 SLLM 的推理能力。

表 5　不同数据集中组合模型的有效性对比　　(单位：%)

组合方式	GSM8K	MultiArith	SVAMP	CommonsenseQA
LLaMA2：				
＋LLaMA2	25.9	66.1	42.0	48.7
＋Baichuan2	26.8	68.9	43.0	57.0
＋ChatGLM3	27.1	70.6	45.3	58.6
Baichuan2：				
＋LLaMA2	32.7	76.1	44.0	46.7
＋Baichuan2	33.5	82.6	53.7	59.1
＋ChatGLM3	34.4	83.9	53.7	59.5
ChatGLM3：				
＋LLaMA2	53.7	88.9	61.3	54.5
＋Baichuan2	53.7	88.9	63.3	60.9
＋ChatGLM3	54.3	90.6	64.3	63.7

注：下划线表示 SLLM.I，＋表示 SLLM.II。

4　结论

本文提出了一个基于 SLLM 的问题推理框架，通过在推理路径生成和答案生成两个阶段之间引入基于逐步语义验证器(SSVRP)的推理路径排序选择机制，在无人干预情况下实现 SLLM 推理能力提升。相关实验结果表明，推理路径的优劣直接影响 SLLM 的推理结果，有效的推理路径排序与选择方法有助于提升 SLLM 的推理能力。利用多个(两个)SLLM 的协作方法能够使它们获得与单个 LLM 相当的推理能力。语义验证器能够通过选择最优推理路径来增强 SLLM 的推理能力，在推理路径评价过程中，采用推理步骤进行语义验证的 SSVRP 方法优于采用推理路径进行语义验证的 SVRP 方法。

参考文献

[1] OpenAI. ChatGPT：A large-scale generative model for open-domain chat. [CP/OL]. [2021-09-11]. https://chat.openai.com/

[2] Google.Gemini [CP/OL]. [2024-07-11]. https://gemini.google.com.

[3] TOUVRON H, MARTIN L, STONE K, et al. LLaMA 2：Open foundation and fine-tuned chat

models[J]. arXiv preprint arXiv: 2307.09288, 2023.
[4] ROBINSON R B, JOHANSSON K, FEY J C, et al. Edu-larp @ CHI [C]//Proceedings of the CHI Conference on Human Factors in Computing Systems, 2023: 1-5.
[5] YANG W, LI C, ZHANG J, et al. Bigtranslate: Augmenting large language models with multilingual translation capability over 100 languages [J]. arXiv preprint arXiv: 2305.18098, 2023.
[6] SIMMERING P F, HUOVIALA P. Large language models for aspect-based sentiment analysis[J]. arXiv preprint arXiv: 2310.18025, 2023.
[7] LU A, ZHANG H, ZHANG Y, et al. Bounding the capabilities of large language models in open text generation with prompt constraints[J]. arXiv preprint arXiv: 2302.09185, 2023.
[8] BARLA N. Deploying large NLP models: Infrastructure cost optimization[EB/OL]. [2024-07-11].https://neptune.ai/blog/nlp-models-infrasturcture-cost-optimizution.
[9] Computer hardware required to run LLaMA AI model locally (gpu, cpu, ram, ssd). Allan Witt.[EB/OL]. [2024-04-08]. https://www.hardware-corner.net/guides/computer-to-run-llama-ai-model/
[10] HO N, SCHMID L, YUN S Y. Large language models are reasoning teachers [J]. arXiv preprint arXiv: 2212.10071, 2022.
[11] MAGISTER L C, MALLINSON J, ADAMEK J, et al. Teaching small language models to reason [J]. arXiv preprint arXiv: 2212.08410, 2022.
[12] FU X Y, LASKAR M T R, KHASANOVA E, et al. Tiny titans: Can smaller large language models punch above their weight in the real world for meeting summarization? [J]. arXiv preprint arXiv: 2402.00841, 2024.
[13] WEI J, WANG X, SCHUURMANS D, et al. Chain-of-thought prompting elicits reasoning in large language models [C]//Proceedings of the 36th International Conference on Neural Information Processing Systems, 2022, 35: 24824-24837.
[14] KOJIMA T, GU S S, REID M, et al. Large language models are zero-shot reasoners[C]//Proceedings of the 36th International Conference on Neural Information Processing Systems, 2022, 35: 22199-22213.
[15] ZHANG Z, ZHANG A, LI M, et al. Automatic chain of thought prompting in large language models [J]. arXiv preprint arXiv: 2210.03493, 2022.
[16] WANG X, WEI J, SCHUURMANS D, et al. Self-consistency improves chain of thought reasoning in language models [J]. arXiv preprint arXiv: 2203.11171, 2022.
[17] LI Y, LIN Z, ZHANG S, et al. Making language models better reasoners with step-aware verifier [C]//Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 2023: 5315-5333.
[18] FU Y, PENG H, OU L, et al. Specializing smaller language models towards multi-step reasoning[C]//Proceedings of the International Conference on Machine Learning. PMLR, 2023: 10421-10430.
[19] PRESS O, ZHANG M, MIN S, et al. Measuring and narrowing the compositionality gap in language models[J]. arXiv preprint arXiv: 2210.03350, 2022.
[20] WANG B, DENG X, SUN H. Iteratively prompt pre-trained language models for chain of thought[J]. arXiv preprint arXiv: 2203.08383, 2022.
[21] ZHOU D, SCHäRLI N, HOU L, et al. Least-to-most prompting enables complex reasoning in large language models [J]. arXiv preprint arXiv: 2205.10625, 2022.
[22] CHUNG H W, HOU L, LONGPRE S, et al. Scaling instruction-finetuned language models [J]. Journal of Machine Learning Research, 2024, 25(70): 1-53.
[23] ZELIKMAN E, WU Y, MU J, et al. Star: Bootstrapping reasoning with reasoning [C]//Proceedings of the 36th International Conference on Neural Information Processing Systems, 2022, 35: 15476-15488.
[24] MUKHERJEE S, MITRA A, JAWAHAR G, et al. Orca: Progressive learning from complex explanation traces of GPT-4 [J]. arXiv preprint arXiv: 2306.02707, 2023.
[25] GAO T, YAO X, CHEN D. Simcse: Simple contrastive learning of sentence embeddings[J]. arXiv preprint arXiv: 2104.08821, 2021.
[26] HE P, LIU X, GAO J, et al. DeBERTa: Decoding-enhanced bert with disentangled attention[J]. arXiv preprint arXiv: 2006.03654, 2020.
[27] COBBE K, KOSARAJU V, BAVARIAN M, et al. Training verifiers to solve math word problems[J]. arXiv preprint arXiv: 2110.14168, 2021.
[28] ROY S, ROTH D. Solving general arithmetic word problems [J]. arXiv preprint arXiv: 1608.01413, 2016.
[29] PATEL A, BHATTAMISHRA S, GOYAL N. Are NLP models really able to solve simple math word problems? [J]. arXiv preprint arXiv: 2103.07191, 2021.
[30] TALMOR A, HERZIG J, LOURIE N, et al. Commonsenseqa: A question answering challenge targeting commonsense knowledge [J]. arXiv preprint arXiv: 1811.00937, 2018.
[31] BROWN T B. Language models are few-shot learners [J]. arXiv preprint arXiv: 2005.14165, 2020.
[32] CHUNG H W, HOU L, LONGPRE S, et al. Scaling instruction-finetuned language models [J]. Journal of Machine Learning Research, 2024, 25(70): 1-53.
[33] YANG A, XIAO B, WANG B, et al. Baichuan 2:

Open large-scale language models[J]. arXiv preprint arXiv: 2309.10305, 2023.
[34] ZENG A, LIU X, DU Z, et al. GLM-130b: An open bilingual pre-trained model[J]. arXiv preprint arXiv: 2210.02414, 2022.

李俊(1999—)，硕士，主要研究领域为自然语言处理。
E-mail: lijun1@stu.sau.edu.cn

刘雨婷(2002—)，硕士，主要研究领域为自然语言处理。
E-mail: liuyuting4@stu.sau.edu.cn

白宇(1982—)，通信作者，副教授，主要研究领域为自然语言处理、信息检索、语言与知识计算。
E-mail: baiyu@sau.edu.cn

2026 全国知识图谱与语义计算大会(CCKS2026)评测任务征集通知

全国知识图谱与语义计算大会(China Conference on Knowledge Graph and Semantic Computing, CCKS)由中国中文信息学会语言与知识计算专业委员会主办，大会源自中文知识图谱研讨会(Chinese Knowledge Graph Symposium, CKGS)和中国语义网与万维网科学大会(Chinese Semantic Web and Web Science Conference, CSWS)，2016 年两会合并。CCKS 2016、2017、2018、2019、2020、2021、2022、2023、2024、2025 分别在北京、成都、天津、杭州、南昌、广州(线上)、秦皇岛、沈阳、重庆、福州举办。大会已成为国内知识图谱、语义技术等领域的核心学术会议，聚集知识表示与推理、自然语言理解与知识获取、图数据管理与图计算、智能问答等方向的学者与研发人员。

CCKS 2025 评测竞赛环节共包含 8 项竞赛，涵盖知识编辑、知识抽取、复杂问答等多项任务，吸引了超过 2100 支队伍、近 3000 人参赛，共计 15 万奖金，在工业界和学术界形成较高影响力。

本届全国知识图谱与语义计算大会将于 2026 年 8 月 21 日至 23 日在西安召开。本届大会主题为**“知识、记忆与认知推理”**，旨在探讨知识记忆机制与认知推理之间的深度融合与协同演进。大会将聚焦知识表示、知识存储、知识挖掘、知识融合、知识推理、可解释性、记忆增强、认知计算等知识图谱与大模型关键技术，引导知识驱动的新一代认知智能理论技术突破与产业应用发展。大会议程包括讲习班、大会特邀报告、前沿趋势论坛、工业界论坛、青年学者论坛、评测与竞赛、论文报告、海报与系统展示等环节，促进产学研合作。

CCKS 2026 将组织知识图谱与语义计算相关评测竞赛，旨在为研究者提供测试技术、算法及系统的平台，推动知识图谱与大模型结合方向的研究进展与产业落地。CCKS 2026 的评测任务可由任务发布者根据任务特点自行选择平台与评测方案。现面向相关领域研究者、科研机构及企业，公开征集 CCKS 2026 评测任务方案。

会议时间：2026 年 8 月 21－23 日　　**会议地点：西安**

会议官网：http://sigkg.cn/ccks2026/

评测主席：

- 刘井平，中山大学(liujp68@mail.sysu.edu.cn)
- 毕胜，东南大学(bisheng@seu.edu.cn)

评测任务方案请通过邮件发送给评测主席，方案中应详细描述任务内容以及评测数据的准备过程。评测方案的模板及内容可参考 CCKS2025 各个任务的任务描述文件(https://sigkg.cn/ccks2025/evaluation-2/)。

(中国中文信息学会)