

# 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<sup>1</sup>, 白宇<sup>\*1,2</sup>, 刘雨婷<sup>1</sup>

<sup>1</sup>沈阳航空航天大学计算机学院, 辽宁沈阳, 110136

<sup>2</sup>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

lijun1@stu.sau.edu.cn; baiyu@sau.edu.cn; liuyuting4@stu.sau.edu.cn

## 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, 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, 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。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

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

Jun Li<sup>1</sup>, Yu Bai<sup>\*1,2</sup>, Yuting Liu<sup>1</sup>

<sup>1</sup>School of Computer Science, Shenyang Aerospace University,  
Shenyang, Liaoning, 110136

<sup>2</sup>National & Local Joint Engineering Laboratory of Multi-Language  
Collaborative Translation Technology, Shenyang, Liaoning, 110136

lijun1@stu.sau.edu.cn; baiyu@sau.edu.cn; liuyuting4@stu.sau.edu.cn

## Abstract

Although Large Language Models (LLM) have achieved great success in the field of natural language processing, their training with hundreds of billions of parameters also incurs huge computational costs. 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. Although prompting methods such as In-Context Learning (ICL) have improved the problem-handling capabilities of SLLM to a certain extent, manually constructed prompts often require participants to have specific professional domain knowledge, which poses a challenge to the universal promotion of LLM. To address the above issues, this paper proposes a question reasoning framework based on 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

\* 通讯作者 Corresponding author.

of SLLM is improved without human intervention. Experimental results show that SSVRP effectively enhances the reasoning performance of SLLM, with average accuracy rates of 54.3%, 90.6%, 64.3% and 63.7% in four reasoning tasks, respectively, and achieves 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 Paths Ranking

## 1 引言

大型语言模型(Large Language Models, LLM), 如OpenAI的ChatGPT(OpenAI, 2022)、谷歌的Bard(Manyika and Hsiao, 2023)以及Meta的LLaMA(Touvron et al., 2023)等, 在推理问答(Robinson et al., 2023)、语言翻译(Yang et al., 2023b)、情感分析(Simmering and Huoviala, 2023)和文本生成(Lu et al., 2023)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(Barla, 2023)。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6GB VRAM的GPU, 而要运行LLaMA-65B或70B模型, 则至少需要40GB VRAM的GPU(Witt, 2024)。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。

LLaMA模型系列	最小VRAM要求	价格(美元)
LLaMA/Llama-2 7B	6GB	280
LLaMA/Llama-2 13B	10GB	300
LLaMA/Llama-2 33B	20GB	2800
LLaMA/Llama-2 65B/70B	40GB	5600

表 1. 4-Bit量化的LLaMA模型对GPU的需求和最低部署成本(Witt, 2024)

近期的工作(Ho et al., 2022; Magister et al., 2022)表明, 小规模大语言模型(Small-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, 需要更少的内存和存储, 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的参数量的降低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(Fu et al., 2024)。

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需参数调整的情况下预测出准确的输出。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(2022)等人提出思维链(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。

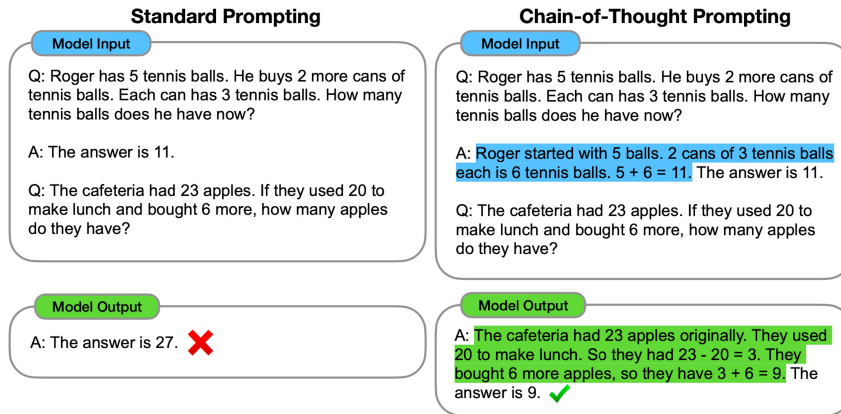


图 1. 标准的输入-输出提示和CoT提示(Wei et al., 2022)

在实际应用中，与使用“Let’s think step by step.”提示的Zero-Shot-CoT方法(Kojima et al., 2022)相比，通过人工手动构建的Manual-CoT(Wei et al., 2022)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(2022)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(2023)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(2023)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。

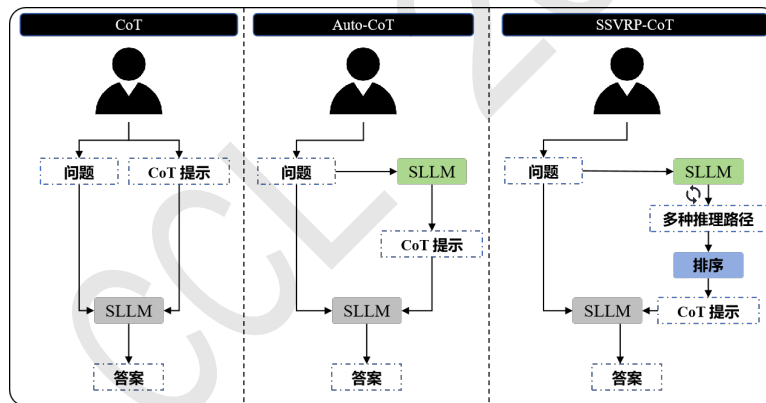


图 2. 本文方法与其它Auto-CoT方法

在本研究中，我们探索在无需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（如图2所示），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。

## 2 相关工作

Wei等人(2022)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(2023)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受不同的提示。Self-Ask(Press et al., 2022)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(Wang et al., 2022)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(Zhou et al., 2022)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到正在进行的上下文中。

Chung等人(2022a)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(Zelikman et al., 2022)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(Fu et al., 2023)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(Mukherjee et al., 2023)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。

## 3 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。

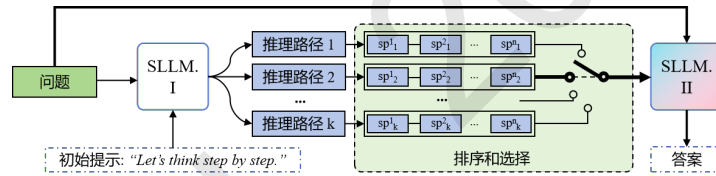


图 3. 基于推理路径排序的问题推理

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。

### 3.1 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 $\mathbf{q}$ ，使用“.”将其拆分为多条信息，记作 $\mathbf{q} = [q_1, \dots, q_p, \dots, q_n]$ ，其中， $q_p$ 表示问题 $\mathbf{q}$ 中的第 $p$ 条信息。假设SLLM-I为问题 $\mathbf{q}$ 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 $\mathbf{R} = [sp_1^{(1)}, \dots, sp_{m_1}^{(1)}, \dots, sp_j^{(i)}, \dots, sp_1^{(k)}, \dots, sp_{m_k}^{(k)}]$ ， $sp_j^{(i)}$ 表示第 $i$ 条推理路径中的第 $j$ 个推理步骤， $i = 1, \dots, k$ ， $j = 1, \dots, m_i$ 。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 $\mathbf{R}$ 中的位置表示为 $Pos^*$ 。

$$Pos^* = \underset{k}{argmax} \left[ \sum_{p=1}^n \sum_{j=1}^{m_k} sim(q_p, sp_j^{(k)}) / m_k \right], \quad (1)$$

其中，函数 $sim(\cdot)$ 的作用是预测问题 $\mathbf{q}$ 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。

### 3.2 逐步语义验证器

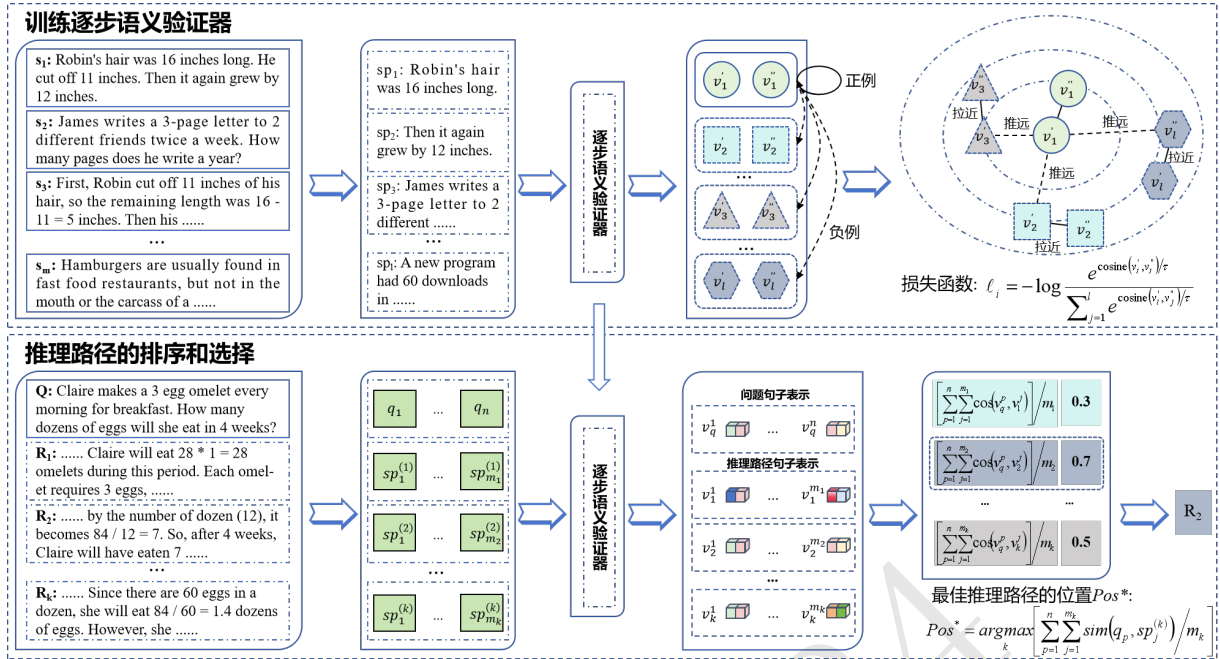


图 4. 逐步语义验证器的训练和用于最佳推理路径排序和选择的过程

为了实现函数 $\operatorname{sim}(\cdot)$ ，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 $v_q^p$ 表示问题 $q$ 的第 $p$ 个信息编码后的表示向量， $v_i^j$ 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 $v_q^p$ 和 $v_i^j$ 之间夹角的余弦值实现函数 $\operatorname{sim}(\cdot)$ 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(Gao et al., 2021)中无监督的对比学习框架，并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 $\{sp_j\}_{j=1}^l$ ，采用dropout噪声和 $sp_j^+ = sp_j$ 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。

$$v_j^z = f_\theta(sp_j, z), \quad (2)$$

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以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：

$$\ell_i = -\log \frac{e^{\cosine(v'_i, v''_i)/\tau}}{\sum_{j=1}^l e^{\cosine(v'_i, v''_j)/\tau}}, \quad (3)$$

其中 $\tau$ 是温度超参数， $\cosine(v', v'')$ 是余弦相似度 $\frac{v'^\top v''}{\|v'\| \|v''\|}$ 。我们使用目标函数 $\ell_i$ 对预训练语言模型DeBERTa-base(He et al., 2021)进行微调，并利用微调后的模型对输入句子进行编码，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。

**算法 1** 推理路径的排序和选择

**Input:** A question:  $q$ , Number of Reason paths generated by *SLLM-I*:  $k$ .

**Output:** The Optimal Position in the List of Reasoning Paths  $Pos^*$ .

```

1: Initial the prompt:  $P \leftarrow$  "Let's think step by step."
2: Initial the Optimal position:  $Pos^* \leftarrow 0$ 
3: Initial the Optimal value:  $Score^* \leftarrow -1$ 
4:  $S[] \leftarrow [Q, SLLM-I(Q, P, k)]$  //SLLM-I生成问题的推理路径
5:  $q[] \leftarrow$  Split  $Q$  by "."
6: for each reasoning path  $R_i$  in  $S/Q$  do
7:    $R_i[] \leftarrow$  Split  $R_i$  by "."
8:    $Score_i \leftarrow 0$ 
9:   for each sentence  $q_p$  in  $q$  do
10:     $Score_{q_p} \leftarrow 0$ 
11:     $v_q^p \leftarrow Semantic\_Verifier(q_p)$ 
    //计算此个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
12:    for each sentence  $sp_j^{(i)}$  in  $R_i$  do
13:       $n \leftarrow 0$ 
14:       $v_i^j \leftarrow Semantic\_Verifier(sp_j^{(i)})$ 
15:       $Score_{sp_j} \leftarrow cosine(v_q^p, v_i^j)$ 
16:       $Score_{q_p} \leftarrow Score_{q_p} + Score_{sp_j}$ 
17:    end for
18:     $Score_i \leftarrow Score_{q_i} + Score_i$ 
19:  end for
20:   $Score_i \leftarrow Score_i/n$ 
  //找出最佳推理路径的位置
21:  if  $Score_i > Score^*$  then
22:     $Score^* \leftarrow Score_i$ 
23:     $Pos^* \leftarrow i$ 
24:  end if
25: end for
26: return  $Pos^*$ 

```

## 4 实验

### 4.1 实验步骤

#### 4.1.1 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，GS-M8K(Cobbe et al., 2021)、MultiArith(Roy and Roth, 2015)以及SVAMP(Patel et al., 2021)3个数据集主要用来评估SLLM解决复杂的多步算术问题的能力，CommonsenseQA(Talmor et al., 2019)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。

#### 4.1.2 训练数据

本文使用的训练数据集包含8593个算术推理问题和9741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77257个算术推理问题的推理路径和87669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。

#### 4.1.3 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

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	7473	1319
MultiArith	数学应用题	420	180
SVAMP	数学应用题	700	300
CommonsenseQA	常识单选题	9741	1221

表 2. 实验数据的详细信息

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：

- LLM

**GPT-3**(Brown et al., 2020)拥有1750亿的参数，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。在本文中，我们引用了Li等人(2023)报告的两个公共引擎davinci和code-davinci-002的实验结果。

- SLLM

**FlanT5-XL(3B)**(Chung et al., 2022b)是T5模型的一种，XL版本的FlanT5模型具有30亿的数量。

**LLaMA2(7B)**(Touvron et al., 2023)是一个经过预训练和微调的大型语言模型，具有70亿的参数。它是Meta开发的LLaMA2的最小版本。

**Baichuan2(7B)**(Yang et al., 2023a)是一个包含70亿的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)**(Zeng et al., 2022)是一个轻量级开源预训练对话模型，包含60亿的参数。它是ChatGLM3的最小版本，ChatGLM3是中英文训练的双语大语言模型，属于智谱AI开发的大型语言模型系列。

- 提示增强方法

**Greedy Decode**(Kojima et al., 2022)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**(Wang et al., 2023)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**(Li et al., 2023)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。

#### 4.1.4 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，其它超参数设置如表3所示。

## 4.2 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。

## 4.3 实验分析

### 4.3.1 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。

模型	超参数	取值			
		SLLM-I	True	SLLM-II	False
SLLM	do_sample		True		False
	top_k		10		1
	temperature		0.7		0
	k		3		-
DeBERTa	dropout		0.1		
	num_heads		8		
	$\tau$		0.05		
	learning_rate		2e-5		
	batch_size		16		
	max_sequence_length		256		

表 3. SLLM和DeBERTa的超参数设置

Method	GSM8K	MultiArith	SVAMP	CommonsenseQA	Average
<u>GPT-3 (davinci, 175B):</u>					
Greedy Decode	8.7	31.4	21.2	48.2	27.4
Self-Consistency	18.9	68.6	44.6	57.4	47.4
<u>GPT-3 (text-davinci-002, 175B):</u>					
Greedy Decode	37.1	70.7	60.0	65.5	58.3
Self-Consistency	58.2	88.4	78.2	72.9	74.4
<u>FlanT5-XL (3B):</u>					
Greedy Decode	5.4	10.0	8.7	77.8	25.5
Self-Consistency	2.9	6.1	7.3	<b>81.9</b>	24.6
DIVERSE (LLM)	<b>6.4</b>	<u>14.4</u>	<u>11.3</u>	79.3	<u>27.9</u>
SSVRP (SLLM)	<u>6.2</u>	<b>16.7</b>	<b>14.0</b>	<u>80.6</u>	<b>29.4</b>
<u>LLaMA2 (7B):</u>					
Greedy Decode	<u>24.5</u>	68.3	39.7	43.7	44.1
Self-Consistency	22.9	<u>68.9</u>	41.3	43.8	44.2
DIVERSE (LLM)	22.8	<b>69.7</b>	<b>44.3</b>	<u>45.6</u>	<u>45.5</u>
SSVRP (SLLM)	<b>25.9</b>	66.1	<u>42.0</u>	<b>48.7</b>	<b>45.7</b>
<u>Baichuan2 (7B):</u>					
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)	<b>34.1</b>	80.6	<b>54.7</b>	58.5	56.9
SSVRP (SLLM)	<u>33.5</u>	<b>82.6</b>	<u>53.7</u>	<b>59.1</b>	<b>57.2</b>
<u>ChatGLM3 (6B):</u>					
Greedy Decode	<u>52.9</u>	86.7	59.3	60.8	64.9
Self-Consistency	45.8	<u>93.9</u>	60.0	61.4	65.3
DIVERSE (LLM)	52.7	<b>94.6</b>	<u>62.3</u>	<u>62.7</u>	<u>68.1</u>
SSVRP (SLLM)	<b>54.3</b>	90.6	<b>64.3</b>	<b>63.7</b>	<b>68.2</b>

表 4. SLLM在不同的推理任务中的有效性比较。最好结果用黑体表示，第二好的结果用下划线表示。

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。

#### 4.3.2 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(7-



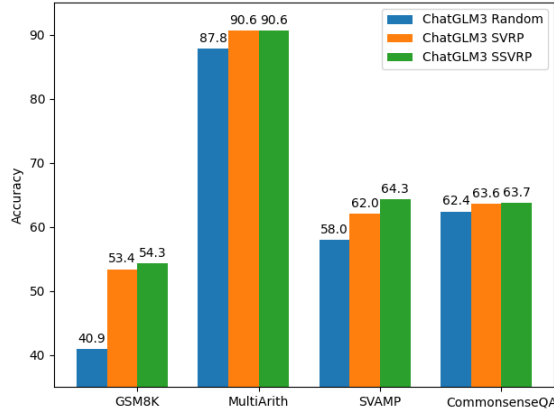


图 5. 在ChatGLM3上的推理路径选择方法的比较

B)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(7-B)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, 我们进行了SLLM-I和SLLM-II的组合实验, 实验结果如表5所示。

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

表 5. 不同数据集中组合模型的有效性对比, 下划线表示SLLM-I, +表示SLLM-II

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。

## 5 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。

## 致谢

本文受到国家自然科学基金(U1908216)的资助。

## 参考文献

- Nilesh Barla. 2023. Deploying large nlp models: Infrastructure cost optimization. Accessed on April 8th, 2024.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022a. Scaling instruction-finetuned language models. *arXiv e-prints*, pages arXiv–2210.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022b. Scaling instruction-finetuned language models.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. *arXiv preprint arXiv:2301.12726*.
- Xue-Yong Fu, Md Tahmid Rahman Laskar, Elena Khasanova, Cheng Chen, and TN ShashiBhushan. 2024. Tiny titans: Can smaller large language models punch above their weight in the real world for meeting summarization? *ArXiv*, abs/2402.00841.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.
- Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022. Large language models are reasoning teachers. *arXiv e-prints*, pages arXiv–2212.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.
- Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making language models better reasoners with step-aware verifier. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada, July. Association for Computational Linguistics.
- Albert Lu, Hongxin Zhang, Yanzhe Zhang, Xuezhi Wang, and Diyi Yang. 2023. Bounding the capabilities of large language models in open text generation with prompt constraints. In Andreas Vlachos and Isabelle Augenstein, editors, *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1982–2008, Dubrovnik, Croatia, May. Association for Computational Linguistics.
- Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2022. Teaching small language models to reason. *arXiv preprint arXiv:2212.08410*.

- James Manyika and Sissie Hsiao. 2023. An overview of bard: an early experiment with generative ai. *AI. Google Static Documents*, 2.
- Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.
- OpenAI. 2022. Chatgpt: A large-scale generative model for open-domain chat. <https://chat.openai.com/>.
- Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *ArXiv*, abs/2210.03350.
- Raquel B Robinson, Karin Johansson, James Collin Fey, Elena Márquez Segura, Jon Back, Annika Waern, Sarah Lynne Bowman, and Katherine Isbister. 2023. Edu-larp@ chi. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–5.
- Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1743–1752.
- Paul F Simmering and Paavo Huoviala. 2023. Large language models for aspect-based sentiment analysis. *arXiv preprint arXiv:2310.18025*.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Boshi Wang, Xiang Deng, and Huan Sun. 2022. Iteratively prompt pre-trained language models for chain of thought. In *Conference on Empirical Methods in Natural Language Processing*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.
- Allan Witt. 2024. Computer hardware required to run llama ai model locally (gpu, cpu, ram, ssd). Accessed on April 8th, 2024.
- Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023a. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.
- Wen Yang, Chong Li, Jiajun Zhang, and Chengqing Zong. 2023b. Bigtrans: Augmenting large language models with multilingual translation capability over 100 languages. *arXiv preprint arXiv:2305.18098*.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.
- Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.
- Zhuosheng Zhang, Aston Zhang, Mu Li, and Alexander J. Smola. 2022. Automatic chain of thought prompting in large language models. *ArXiv*, abs/2210.03493.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.