Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei **Xuezhi Wang** **Dale Schuurmans** **Maarten Bosma**
Brian Ichter **Fei Xia** **Ed H. Chi** **Quoc V. Le** **Denny Zhou**

Google Research, Brain Team
{jasonwei,dennyzhou}@google.com

Abstract

We explore how generating a *chain of thought*—a series of intermediate reasoning steps—significantly improves the ability of large language models to perform complex reasoning. In particular, we show how such reasoning abilities emerge naturally in sufficiently large language models via a simple method called *chain-of-thought prompting*, where a few chain of thought demonstrations are provided as exemplars in prompting.

Experiments on three large language models show that chain-of-thought prompting improves performance on a range of arithmetic, commonsense, and symbolic reasoning tasks. The empirical gains can be striking. For instance, prompting a PaLM 540B with just eight chain-of-thought exemplars achieves state-of-the-art accuracy on the GSM8K benchmark of math word problems, surpassing even finetuned GPT-3 with a verifier.

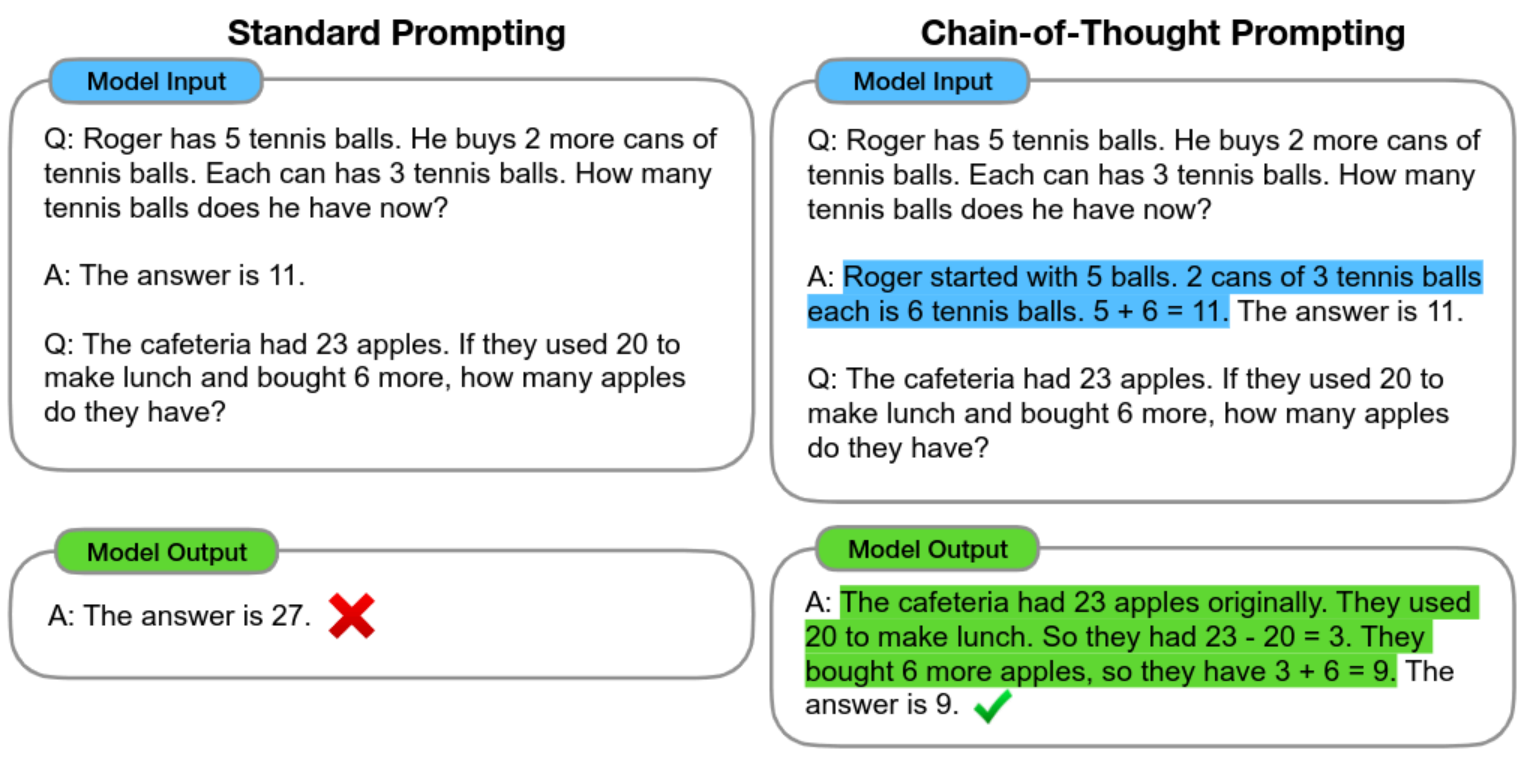


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

大语言模型中的思维链启发式推理

魏杰　王学智　戴尔·舒尔曼斯　马尔滕·博斯马

布莱恩·伊奇特　飞霞　艾德·H 迟　Quoc V. Le　公司简介

谷歌研究，大脑团队
{jasonwei,dennyzhou}@google.com

摘要

我们将探讨如何生成一个思想链-一系列中间推理步骤-显着提高大型语言模型执行复杂推理的能力。特别地，我们展示了这种推理能力是如何在足够大的语言模型中通过一种称为思维链提示的简单方法自然出现的，其中提供了一些思维链示范作为提示中的范例。

在三个大型语言模型上的实验表明，思维链提示提高了一系列算术、常识和符号推理任务的性能。经验上的收获可能是惊人的。例如，仅用 8 个思维链示例提示 PaLM 540 B，就可以在数学应用题的 GSM 8 K 基准测试中实现最先进的准确性，甚至超过了带有验证器的微调 GPT-3。

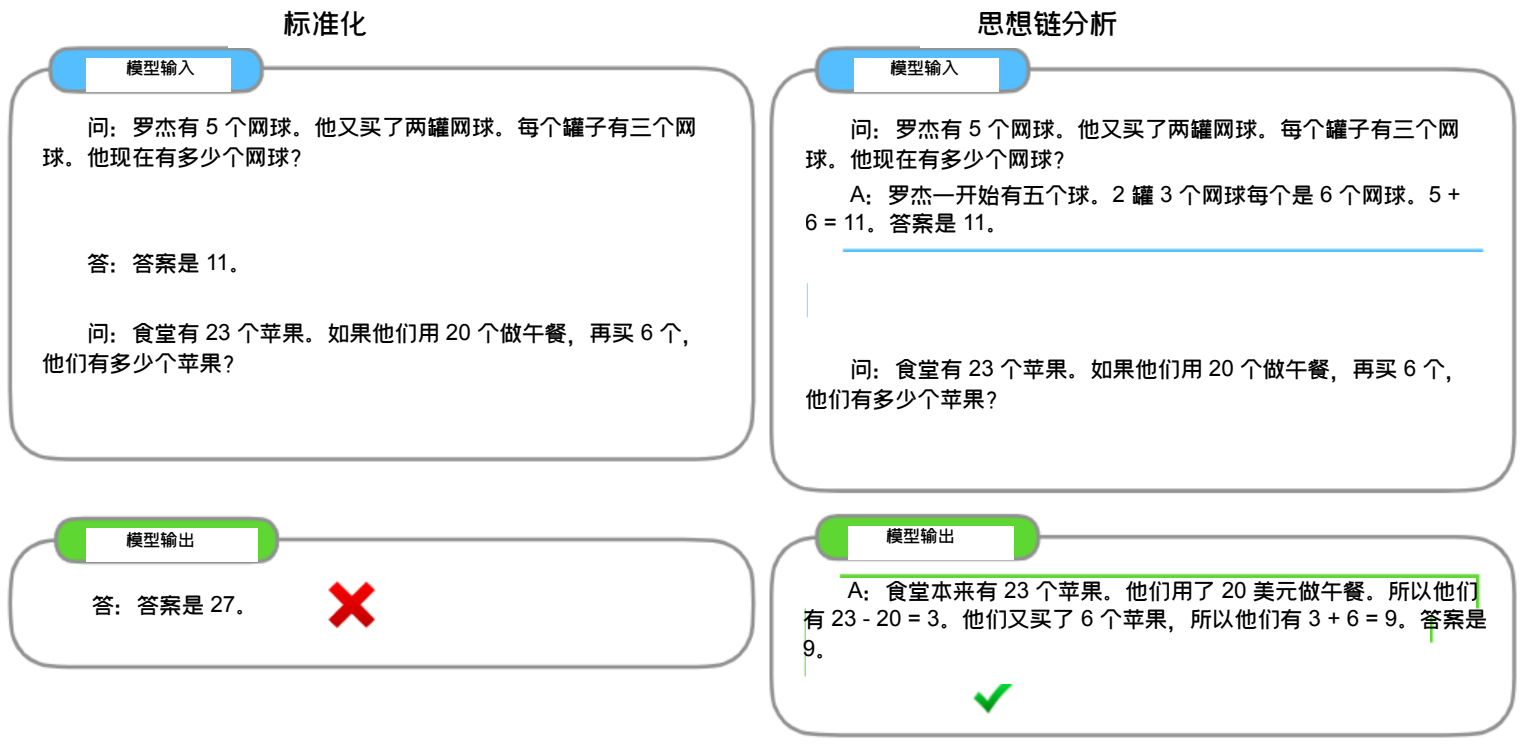


图 1：思维链提示使大型语言模型能够处理复杂的算术、常识和符号推理任务。突出了思维链推理过程。

第 36 届神经信息处理系统会议（NeurIPS 2022）。

1 Introduction

The NLP landscape has recently been revolutionized by language models (Peters et al., 2018; Devlin et al., 2019; Brown et al., 2020, *inter alia*). Scaling up the size of language models has been shown to confer a range of benefits, such as improved performance and sample efficiency (Kaplan et al., 2020; Brown et al., 2020, *inter alia*). However, scaling up model size alone has not proved sufficient for achieving high performance on challenging tasks such as arithmetic, commonsense, and symbolic reasoning (Rae et al., 2021).

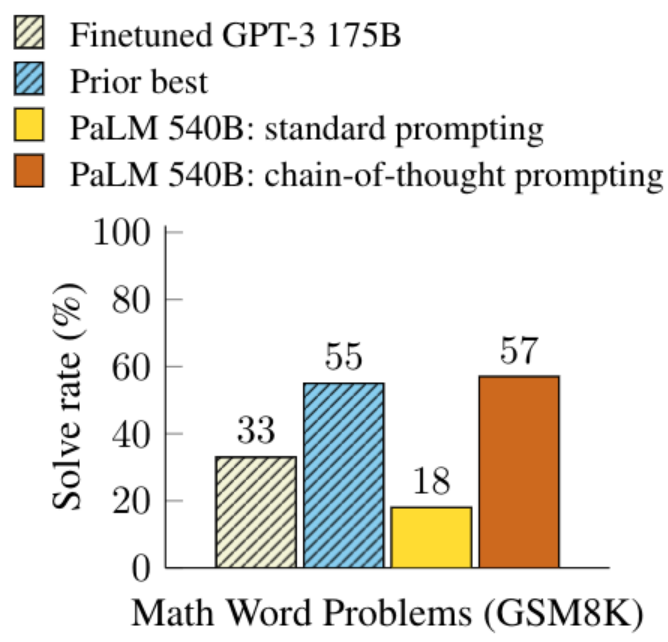


Figure 2: PaLM 540B uses chain-of-thought prompting to achieve new state-of-the-art performance on the GSM8K benchmark of math word problems. Finetuned GPT-3 and prior best are from Cobbe et al. (2021).

This work explores how the reasoning ability of large language models can be unlocked by a simple method motivated by two ideas. First, techniques for arithmetic reasoning can benefit from generating natural language rationales that lead to the final answer. Prior work has given models the ability to generate natural language intermediate steps by training from scratch (Ling et al., 2017) or finetuning a pretrained model (Cobbe et al., 2021), in addition to neuro-symbolic methods that use formal languages instead of natural language (Roy and Roth, 2015; Chiang and Chen, 2019; Amini et al., 2019; Chen et al., 2019). Second, large language models offer the exciting prospect of in-context few-shot learning via *prompting*. That is, instead of finetuning a separate language model checkpoint for each new task, one can simply "prompt" the model with a few input–output exemplars demonstrating the task. Remarkably, this has been successful for a range of simple question-answering tasks (Brown et al., 2020).

Both of the above ideas, however, have key limitations. For rationale-augmented training and finetuning methods, it is costly to create a large set of high quality rationales, which is much more complicated than simple input–output pairs used in normal machine learning. For the traditional few-shot prompting method used in Brown et al. (2020), it works poorly on tasks that require reasoning abilities, and often does not improve substantially with increasing language model scale (Rae et al., 2021). In this paper, we combine the strengths of these two ideas in a way that avoids their limitations. Specifically, we explore the ability of language models to perform few-shot prompting for reasoning tasks, given a prompt that consists of triples: ⟨input, *chain of thought*, output⟩. A *chain of thought* is a series of intermediate natural language reasoning steps that lead to the final output, and we refer to this approach as *chain-of-thought prompting*. An example prompt is shown in Figure 1.

We present empirical evaluations on arithmetic, commonsense, and symbolic reasoning benchmarks, showing that chain-of-thought prompting outperforms standard prompting, sometimes to a striking degree. Figure 2 illustrates one such result—on the GSM8K benchmark of math word problems (Cobbe et al., 2021), chain-of-thought prompting with PaLM 540B outperforms standard prompting by a large margin and achieves new state-of-the-art performance. A prompting only approach is important because it does not require a large training dataset and because a single model checkpoint can perform many tasks without loss of generality. This work underscores how large language models can learn via a few examples with natural language data about the task (c.f. automatically learning the patterns underlying inputs and outputs via a large training dataset).

2 Chain-of-Thought Prompting

Consider one's own thought process when solving a complicated reasoning task such as a multi-step math word problem. It is typical to decompose the problem into intermediate steps and solve each before giving the final answer: *"After Jane gives 2 flowers to her mom she has 10 . . . then after she gives 3 to her dad she will have 7 . . . so the answer is 7."* The goal of this paper is to endow language models with the ability to generate a similar *chain of thought*—a coherent series of intermediate reasoning steps that lead to the final answer for a problem. We will show that sufficiently large

1 引言

NLP 的前景最近已经被语言模型所彻底改变（Peters 等人，2018 年; Devlin 等人，2019 年; Brown 等人，2020 年，阿利亚其他外）。语言模型尺寸的扩大已经显示出带来了一系列的好处，例如改进的性能和样本效率（Kaplan 等人，2020; Brown 等人，2020 年，阿利亚其他外）。然而，仅按比例增大模型大小还不足以在诸如算术、常识和符号推理等具有挑战性的任务上实现高性能（Rae 等人，2021 年）。

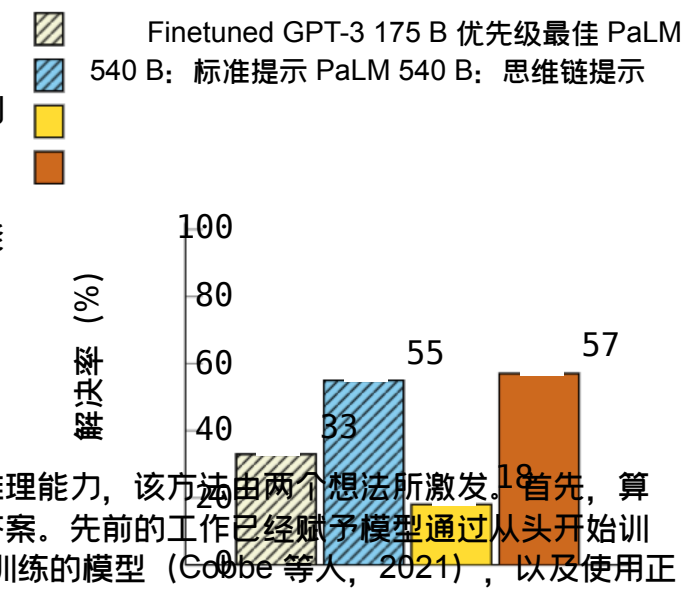


图 2：PaLM 540 B 使用了思想链提示，在数学应用题的 GSM 8 K 基准测试中实现了最先进的性能。微调 GPT-3 和既往最佳值来自 Cobbe et al.（2021）。

本文探讨了如何通过一个简单的方法来解锁大型语言模型的推理能力，该方法由两个想法所激发。首先，算术推理的技巧可以从产生自然语言推理中获益，从而得到最终的答案。先前的工作已经赋予模型通过从头开始训练来生成自然语言中间步骤的能力（Ling 等人，2017）或微调预训练的模型（Cobbe 等人，2021），以及使用正式语言而非自然语言的神经符号方法（Roy 和 Roth，2015; Chiang 和 Chen，2019 年; Chen 等人，2019 年）。第二，大型语言模型提供了通过提示进行上下文少量学习的令人兴奋的前景。也就是说，不必为每个新任务微调一个单独的语言模型检查点，可以简单地用几个演示该任务的输入-输出样本来“提示”模型。值得注意的是，这对于一系列简单的问答任务是成功的（Brown 等人，2020 年）

然而，上述两种想法都有关键的局限性。对于有理扩充训练和微调方法，创建大量高质量的有理集是昂贵的，这比在普通机器学习中使用的简单输入-输出对复杂得多。对于 Brown et al.（2020）中使用的传统的少镜头提示方法，其在需要推理能力的任务上效果不佳，并且通常不会随着语言模型规模的增加而显著改善（Rae et al.，2021 年）。在本文中，我们联合收割机的优势，这两个想法的方式，避免其局限性。具体来说，我们探索语言模型执行推理任务的几次提示的能力，给出了一个由三个组成的提示：输入，思维链，输出。思维链是一系列中间自然语言推理步骤，这些步骤导致最终输出，我们将这种方法称为思维链提示。图 1 中显示了一个示例提示。

我们对算术、常识和符号推理基准进行了实证评估，结果表明，思维链提示优于标准提示，有时甚至达到了惊人的程度。图 2 示出了一个这样的结果-在数学单词问题的 GSM 8 K 基准上（Cobbe 等人，2021 年），PaLM 540 B 的思维链提示功能大大优于标准提示功能，实现了全新的最先进性能。仅提示方法是重要的，因为它不需要大的训练数据集，并且因为单个模型检查点可以执行许多任务而不失一般性。这项工作强调了如何大的语言模型可以学习通过几个例子与自然语言数据的任务（参见。经由大的训练数据集自动学习作为输入和输出的基础的模式）。

2　思想链分析

在解决一个复杂的推理任务时，比如一个多步骤的数学单词问题，要考虑自己的思维过程。通常情况下，我们会把问题分解成几个中间步骤，然后逐一解决，最后才给出答案：“简给了她妈妈两朵花，她就有了 10 朵。. .然后她给了她爸爸 3 个之后她就有 7 个了。. .所以答案是 7。”本文的目标是赋予语言模型以产生类似的思维链的能力--一系列连贯的中间推理步骤，从而得出问题的最终答案。我们将证明足够大

language models can generate chains of thought if demonstrations of chain-of-thought reasoning are provided in the exemplars for few-shot prompting.

Figure 1 shows an example of a model producing a chain of thought to solve a math word problem that it would have otherwise gotten incorrect. The chain of thought in this case resembles a solution and can interpreted as one, but we still opt to call it a chain of thought to better capture the idea that it mimics a step-by-step thought process for arriving at the answer (and also, solutions/explanations typically come *after* the final answer (Narang et al., 2020; Wiegreffe et al., 2022; Lampinen et al., 2022, *inter alia*)).

Chain-of-thought prompting has several attractive properties as an approach for facilitating reasoning in language models.

1. First, chain of thought, in principle, allows models to decompose multi-step problems into intermediate steps, which means that additional computation can be allocated to problems that require more reasoning steps.
2. Second, a chain of thought provides an interpretable window into the behavior of the model, suggesting how it might have arrived at a particular answer and providing opportunities to debug where the reasoning path went wrong (although fully characterizing a model's computations that support an answer remains an open question).
3. Third, chain-of-thought reasoning can be used for tasks such as math word problems, commonsense reasoning, and symbolic manipulation, and is potentially applicable (at least in principle) to any task that humans can solve via language.
4. Finally, chain-of-thought reasoning can be readily elicited in sufficiently large off-the-shelf language models simply by including examples of chain of thought sequences into the exemplars of few-shot prompting.

In empirical experiments, we will observe the utility of chain-of-thought prompting for arithmetic reasoning (Section 3), commonsense reasoning (Section 4), and symbolic reasoning (Section 5).

3 Arithmetic Reasoning

We begin by considering math word problems of the form in Figure 1, which measure the arithmetic reasoning ability of language models. Though simple for humans, arithmetic reasoning is a task where language models often struggle (Hendrycks et al., 2021; Patel et al., 2021, *inter alia*). Strikingly, chain-of-thought prompting when used with the 540B parameter language model performs comparably with task-specific finetuned models on several tasks, even achieving new state of the art on the challenging GSM8K benchmark (Cobbe et al., 2021).

3.1 Experimental Setup

We explore chain-of-thought prompting for various language models on multiple benchmarks.

Benchmarks. We consider the following five math word problem benchmarks: **(1)** the **GSM8K** benchmark of math word problems (Cobbe et al., 2021), **(2)** the **SVAMP** dataset of math word problems with varying structures (Patel et al., 2021), **(3)** the **ASDiv** dataset of diverse math word problems (Miao et al., 2020), **(4)** the **AQuA** dataset of algebraic word problems, and **(5)** the **MAWPS** benchmark (Koncel-Kedziorski et al., 2016). Example problems are given in Appendix Table 12.

Standard prompting. For the baseline, we consider standard few-shot prompting, popularized by Brown et al. (2020), in which a language model is given in-context exemplars of input–output pairs before outputting a prediction for a test-time example. Exemplars are formatted as questions and answers. The model gives the answer directly, as shown in Figure 1 (left).

Chain-of-thought prompting. Our proposed approach is to augment each exemplar in few-shot prompting with a chain of thought for an associated answer, as illustrated in Figure 1 (right). As most of the datasets only have an evaluation split, we manually composed a set of eight few-shot exemplars with chains of thought for prompting—Figure 1 (right) shows one chain of thought exemplar, and the full set of exemplars is given in Appendix Table 20. (These particular exemplars did not undergo prompt engineering; robustness is studied in Section 3.4 and Appendix A.2.) To investigate whether chain-of-thought prompting in this form can successfully elicit successful reasoning across a range of

我们将表明，足够大的语言模型可以产生的思想链，如果示范链的思维推理提供了几杆提示的范例。

图 1 显示了一个模型的示例，该模型生成一个思想链来解决一个数学应用题，否则它将变得不正确。在这种情况下，思维链类似于解决方案，可以解释为解决方案，但我们仍然选择称之为思维链，以更好地捕捉它模仿一步一步的思维过程以获得答案的想法（而且，解决方案/解释通常在最终答案之后出现（Narang 等人，2020; Wiegreffe 等人，2022 年; Lampinen 等人，2022 年，阿利亚其他外））。

作为一种促进语言模型推理的方法，思维链提示有几个有吸引力的特性。

1. 首先，原则上，思维链允许模型将多步骤问题分解为中间步骤，这意味着可以将额外的计算分配给需要更多推理步骤的问题。

2. 其次，思维链为模型的行为提供了一个可解释的窗口，提示它如何得出特定的答案，并提供调试推理路径出错的机会（尽管完全表征支持答案的模型计算仍然是一个悬而未决的问题）。

3. 第三，思维链推理可用于数学应用题、常识推理和符号操作等任务，并且可能适用于（至少在原则上）人类可以通过语言解决的任何任务。

4. 最后，在足够大的现成语言模型中，只要将思维链序列的例子包括到几次提示的例子中，就可以很容易地引出思维链推理。

在实证实验中，我们将观察思想链提示在算术推理（第 3 节）、常识推理（第 4 节）和符号推理（第 5 节）中的效用。

3 算术推理

我们开始考虑图 1 中形式的数学应用题，它衡量语言模型的算术推理能力。尽管算术推理对人类来说很简单，但它是语言模型经常难以完成的任务（Hendrycks 等人，2021 年; Patel 等人，第 2021 号决议）。引人注目的是，当与 540 B 参数语言模型一起使用时，思路链提示在若干任务上与任务特定的微调模型相当地执行，甚至在具有挑战性的 GSM 8 K 基准上实现了新的技术状态（Cobbe 等人，2021 年）。

3.1 实验装置

我们在多个基准上探索了各种语言模型的思路链提示。

基准。我们考虑以下五个数学单词问题基准：（1）数学单词问题的 GSM 8 K 基准（Cobbe 等人，2021），（2）具有不同结构的数学单词问题的 SVAMP 数据集（Patel 等人，2021），（3）不同数学单词问题的 ASDiv 数据集（Miao 等人，2020），（4）代数单词问题的 AQuA 数据集，以及（5）MAWPS 基准（Koncel-Kedziorski 等人，2016 年）。附录表 12 中给出了示例问题。

标准提示。对于基线，我们考虑由 Brown et al.（2020）推广的标准少镜头提示，其中在输出测试时间示例的预测之前，在输入-输出对的上下文样本中给出语言模型。范例的格式为问题和答案。模型直接给出了答案，如图 1（左）所示。

思维链提示。我们提出的方法是，用一个思维链来扩充少镜头提示中的每个样本，以获得相关的答案，如图 1（右）所示。由于大多数数据集只有一个评价分割，我们手动合成了一组 8 个少镜头样本，并带有提示思路链--图 1（右）显示了一个思路链样本，附录表 20 中给出了完整的样本集。（这些特殊样本没有经过及时的工程设计;稳健性研究见第 3.4 节和附录 A.2。）为了研究这种形式的思维链提示是否能成功地在一系列

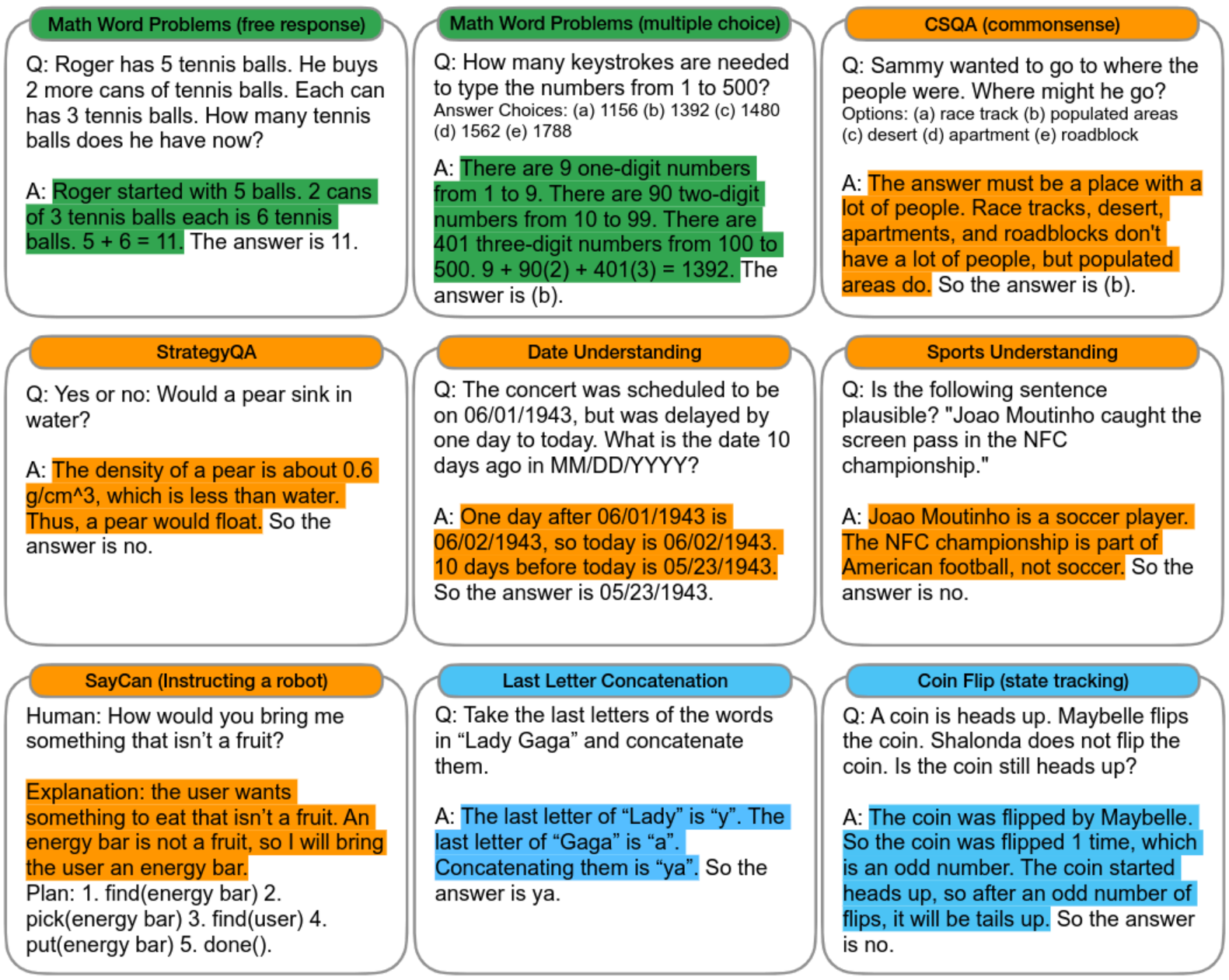


Figure 3: Examples of ⟨input, chain of thought, output⟩ triples for arithmetic, commonsense, and symbolic reasoning benchmarks. Chains of thought are highlighted. Full prompts in Appendix G.

math word problems, we used this single set of eight chain of thought exemplars for all benchmarks except AQuA, which is multiple choice instead of free response. For AQuA, we used four exemplars and solutions from the training set, as given in Appendix Table 21.

Language models. We evaluate five large language models. The first is **GPT-3** (Brown et al., 2020), for which we use text-ada-001, text-babbage-001, text-curie-001, and text-davinci-002, which presumably correspond to InstructGPT models of 350M, 1.3B, 6.7B, and 175B parameters (Ouyang et al., 2022).The second is **LaMDA** (Thoppilan et al., 2022), which has models of 422M, 2B, 8B, 68B, and 137B parameters. The third is **PaLM**, which has models of 8B, 62B, and 540B parameters. The fourth is **UL2 20B** (Tay et al., 2022), and the fifth is **Codex** (Chen et al., 2021, code-davinci-002 in the OpenAI API). We sample from the models via greedy decoding (though follow-up work shows chain-of-thought prompting can be improved by taking the majority final answer over many sampled generations (Wang et al., 2022a)). For LaMDA, we report averaged results over five random seeds, where each seed had a different randomly shuffled order of exemplars. As LaMDA experiments did not show large variance among different seeds, to save compute we report results for a single exemplar order for all other models.

3.2 Results

The strongest results of chain-of-thought prompting are summarized in Figure 4, with all experimental outputs for each model collection, model size, and benchmark shown in Table 2 in the Appendix. There are three key takeaways. First, Figure 4 shows that chain-of-thought prompting is an emergent ability of model scale (Wei et al., 2022b). That is, chain-of-thought prompting does not positively impact performance for small models, and only yields performance gains when used with models of $\sim$100B parameters. We qualitatively found that models of smaller scale produced fluent but illogical chains of thought, leading to lower performance than standard prompting.

数学应用题（免费回答）

问：罗杰有 5 个网球。他又买了两罐网球。每个罐子有三个网球。他现在有多少个网球？

A：罗杰一开始有五个球。2 罐 3 个网球每个是 6 个网球。5 + 6 = 11。答案是 11。

数学应用题（多项选择）

问：输入 1 到 500 之间的数字需要多少次键盘？

答案选择：（a）1156（B）1392（c）1480（d）1562（e）1788

答：从 1 到 9 有 9 个一位数。从 10 到 99 有 90 个两位数。从 100 到 500 有 401 个三位数。9 + 90（2）+ 401（3）= 1392。答案是（B）。

CSQA（常识）

问：萨米想去人们所在的地方。他会去哪里？

选项：（a）赛道（B）居民区（c）沙漠（d）公寓（e）路障

答：答案一定是人多的地方。赛马场、沙漠、公寓和路障没有很多人，但人口稠密的地区有。所以答案是（B）。

战略 QA

问：是或否：一个梨会沉在水里吗？

答：梨的密度约为 0.6 克/立方厘米，比水的密度小。因此，梨会浮起来。所以答案是否定的。

日期理解

问：音乐会原定于 1943 年 6 月 1 日举行，但推迟了一天到今天。10 天前的 MM/DD/YYYY 是什么日期？

答：1943 年 6 月 1 日之后的一天是 1943 年 6 月 2 日，所以今天是 1943 年 6 月 2 日。10 天前的今天是 1943 年 5 月 23 日

所以答案是 1943 年 5 月 23 日。

体育理解

问：下面这句话有道理吗？“穆蒂尼奥在近距离足球锦标赛中接到了屏幕上的传球。"

A：Joao Moutinho 是足球运动员。NFC 锦标赛是美式足球的一部分，而不是足球。所以答案是否定的。

SayCan（指导机器人）

人类：你怎么能给我带来不是水果的东西呢？

解释：用户想要吃的东西不是水果。能量棒不是水果，所以我会给用户带来能量棒。

计划：1.找到（能量棒）2. pick（能量棒）3. find（user）4. 放（能量棒）5. done（）.

最后一个字母连接

问：把“Lady Gaga”中单词的最后一个字母连接起来。

A：“Lady”的最后一个字母是“y”。“Gaga”的最后一个字母是“a”。连接它们的是“ya”。所以答案是你。

硬币翻转（状态跟踪）

问：硬币是正面朝上的。梅贝尔抛硬币。沙隆达没有抛硬币。硬币还是两人对决赛吗？

答：硬币是由梅贝尔抛的。所以硬币被抛了 1 次，这是一个奇数。硬币开始时是正面朝上的，所以在奇数次翻转后，它将是反面朝上的。所以答案是否定的。

为了研究这种形式的思维链提示是否可以成功地在一系列图 3 的范围内引发成功的推理：算术，常识和符号推理基准的输入，思维链，输出三元组的示例。思想的链条被凸显出来。完整提示见附录 G。

数学单词问题，我们在所有基准测试中使用了这一套八个思维链示例，除了 AQuA，它是多项选择而不是自由回答。对于 AQuA，我们使用了培训集中的四个样本和解决方案，如附录表 21 所示。

语言模型。我们评估了五个大型语言模型。第一种是 GPT-3（Brown 等，2020），对此我们使用 text-ada-001、text-babbage-001、text-curie-001 和 text-davinci-002，它们大概对应于 350 M、1.3B、6.7B 和 175 B 参数的 InstructGPT 模型（Ouyang 等人，2022).第二种是 LaMDA（Thoppilan 等人，2022 年），其具有 422 M、2B、8B、68 B 和 137 B 参数的型号。第三种是 PaLM，它有 8B、62 B 和 540 B 参数的模型。第四种是 UL 2 20 B（Tay 等人，2022），第五个是 Codex（Chen 等人，2021，OpenAI API 中的 code-davinci-002）。我们通过贪婪解码从模型中采样（尽管后续工作表明，通过在许多采样代上取多数最终答案，可以改进思维链提示（Wang 等人，2022 a））的规定。对于 LaMDA，我们报告了五个随机种子的平均结果，其中每个种子具有不同的样本随机混合顺序。 由于 LaMDA 实验在不同种子之间没有显示出大的差异，为了节省计算，我们报告了所有其他模型的单个样本顺序的结果。

3.2 结果

图 4 总结了思路链提示的最强结果，附录中的表 2 显示了每个模型集合、模型大小和基准的所有实验输出。这里有三个关键要点。首先，图 4 显示了思维链提示是模型规模的一种突现能力（Wei 等人，第 2022 条 b 款）。也就是说，思路链提示不会对小模型的性能产生积极的影响，并且只有在与约 100B 个参数的模型一起使用时才会产生性能增益。我们定性地发现，规模较小的模型产生了流畅但不合逻辑的思维链，导致了比标准提示更低的表现。

Second, chain-of-thought prompting has larger performance gains for more-complicated problems. For instance, for GSM8K (the dataset with the lowest baseline performance), performance more than doubled for the largest GPT and PaLM models. On the other hand, for SingleOp, the easiest subset of MAWPS which only requires a single step to solve, performance improvements were either negative or very small (see Appendix Table 3).

Third, chain-of-thought prompting via GPT-3 175B and PaLM 540B compares favorably to prior state of the art, which typically finetunes a task-specific model on a labeled training dataset. Figure 4 shows how PaLM 540B uses chain-of-thought prompting to achieve new state of the art on GSM8K, SVAMP, and MAWPS (though note that standard prompting already passed the prior best for SVAMP). On the other two datasets, AQuA and ASDiv, PaLM with chain-of-thought prompting reaches within 2% of the state of the art (Appendix Table 2).

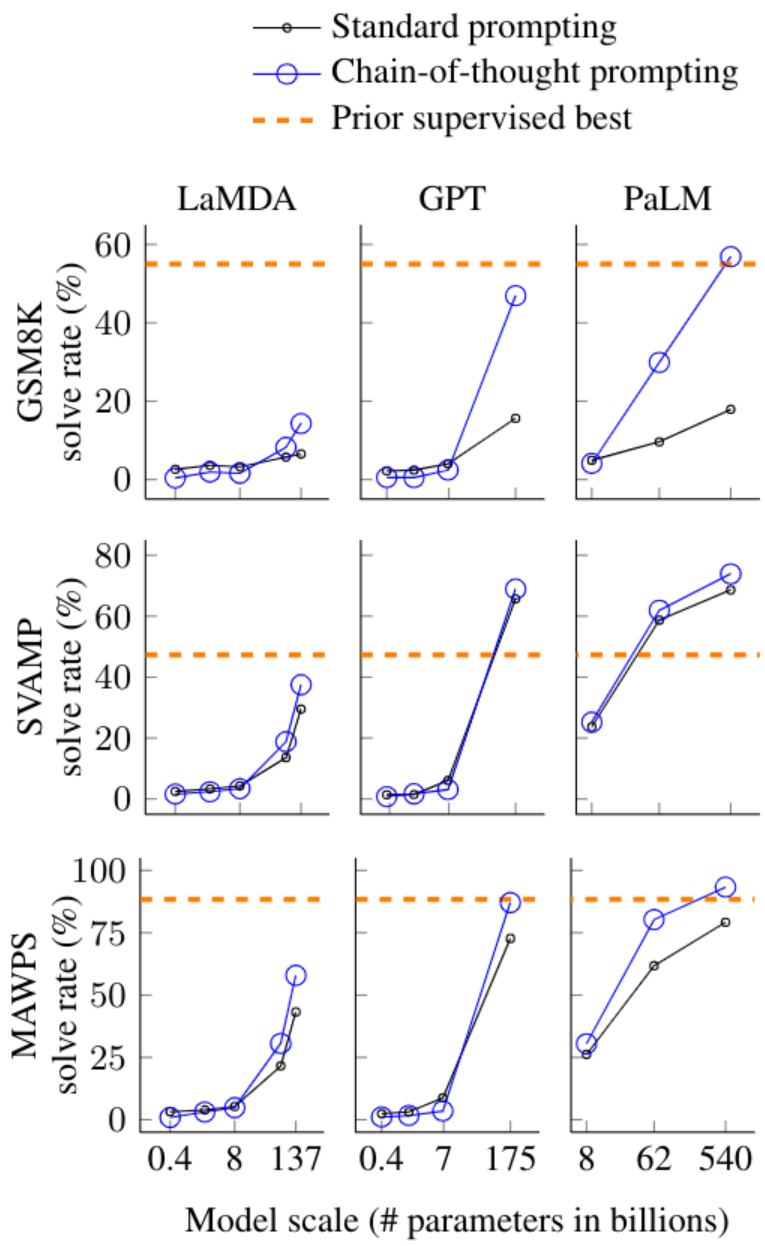


Figure 4: Chain-of-thought prompting enables large language models to solve challenging math problems. Notably, chain-of-thought reasoning is an emergent ability of increasing model scale. Prior best numbers are from Cobbe et al. (2021) for GSM8K, Jie et al. (2022) for SVAMP, and Lan et al. (2021) for MAWPS.

To better understand why chain-of-thought prompting works, we manually examined model-generated chains of thought by LaMDA 137B for GSM8K. Of 50 random examples where the model returned the correct final answer, all of the generated chains of thought were also logically and mathematically correct except two that coincidentally arrived at the correct answer (see Appendix D.1, and Table 8 for examples of correct model-generated chains of thought). We also randomly examined 50 random samples for which the model gave the wrong answer. The summary of this analysis is that 46% of the chains of thought were almost correct, barring minor mistakes (calculator error, symbol mapping error, or one reasoning step missing), and that the other 54% of the chains of thought had major errors in semantic understanding or coherence (see Appendix D.2). To provide a small insight into why scaling improves chain-of-thought reasoning ability, we performed a similar analysis of errors made by PaLM 62B and whether those errors were fixed by scaling to PaLM 540B. The summary is that scaling PaLM to 540B fixes a large portion of one-step missing and semantic understanding errors in the 62B model (see Appendix A.1).

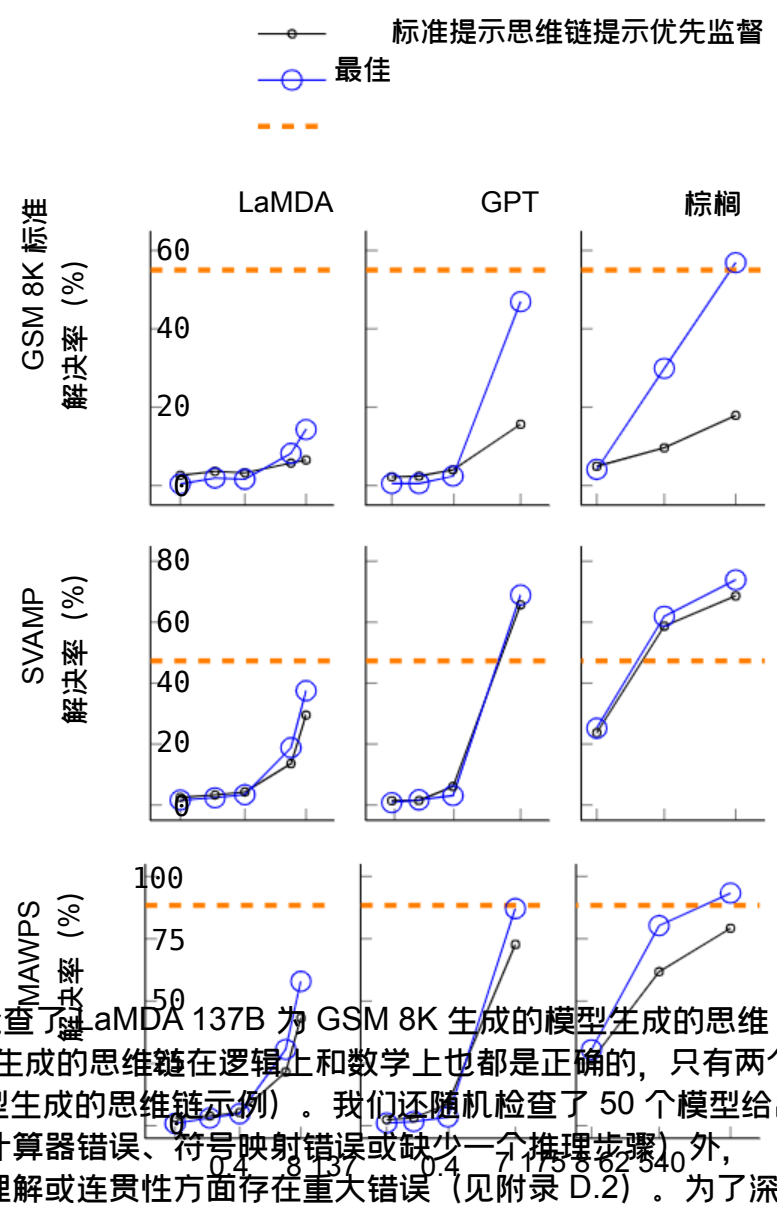
3.3 Ablation Study

The observed benefits of using chain-of-thought prompting raises the natural question of whether the same performance improvements can be conferred via other types of prompting. Figure 5 shows an ablation study with three variations of chain of thought described below.

Equation only. One reason for why chain-of-thought prompting might help is that it produces the mathematical equation to be evaluated, and so we test a variation where the model is prompted to output only a mathematical equation before giving the answer. Figure 5 shows that equation only prompting does not help much for GSM8K, which implies that the semantics of the questions in GSM8K are too challenging to directly translate into an equation without the natural language reasoning steps in chain of thought. For datasets of one-step or two-step problems, however, we find that equation only prompting does improve performance, since the equation can be easily derived from the question (see Appendix Table 6).

其次，对于更复杂的问题，思维链提示有更大的性能增益。例如，对于 GSM8K（基线性能最低的数据集），最大的 GPT 和 PaLM 模型的性能增加了一倍以上。另一方面，对于 SingleOp，最简单的 MAWPS 子集，只需要一个步骤来解决，性能改进要么是负面的，要么非常小（见附录表 3）。

第三，经由 GPT-3 175 B 和 PaLM 540 B 的思维链提示与现有技术相比是有利的，现有技术通常在标记的训练数据集上微调特定于任务的模型。图 4 显示了 PaLM 540 B 如何使用思想链提示来实现 GSM 8 K、SVAMP 和 MAWPS 的最新技术水平（尽管注意标准提示已经超过了 SVAMP 的最佳水平）。在其他两个数据集（AQuA 和 ASDiv）上，具有思维链提示的 PaLM 达到最新技术水平的 2%以内（附录表 2）。



为了更好地理解为什么思维链提示有效，我们手动检查了 LaMDA 137B 为 GSM 8K 生成的模型生成的思维链。在模型返回正确最终答案的 50 个随机示例中，所有生成的思维链在逻辑上和数学上也都是正确的，只有两个碰巧得到了正确答案（参见附录 D.1 和表 8 中的正确模型生成的思维链示例）。我们还随机检查了 50 个模型给出错误答案的随机样本。 该分析的总结是，除了小错误（计算器错误、符号映射错误或缺少一个推理步骤）外，46%的思维链几乎是正确的，另外 54%的思维链在语义理解或连贯性方面存在重大错误（见附录 D.2）。为了深入了解为什么缩放可以提高思维链推理能力，我们对 PaLM 62B 产生的错误进行了更详细的分析，以及其中一些错误是否通过缩放到 PaLM 540 B 而得到修复。总的来说，将 PaLM 扩展到 540 B 修复了 62 B 模型中的大部分一步缺失和语义理解错误（参见附录 A.1）。

图 4：思维链提示使大型语言模型能够解决具有挑战性的数学问题。值得注意的是，思维链推理是一种增加模型规模的新兴能力。之前的最佳数字来自 Cobbe 等人（2021）的 GSM 8K，Jie 等人（2022）的 SVAMP 和 Lan 等人（2021）的 MAWPS。

3.3 消融研究

观察到的使用思维链提示的好处提出了一个自然的问题，即是否可以通过其他类型的提示来赋予相同的性能改进。图 5 显示了一项消融研究，其思路链有三种变化，如下所述。

只有方程式。为什么思维链提示可能会有帮助的一个原因是，它产生了要评估的数学方程，所以我们测试了一个变体，其中模型在给出答案之前被提示只输出一个数学方程。图 5 显示，仅等式提示对 GSM8K 没有太大帮助，这意味着 GSM8K 中问题的语义太具挑战性，无法在思维链中没有自然语言推理步骤的情况下直接翻译成等式。然而，对于一步或两步问题的数据集，我们发现仅提示方程确实可以提高性能，因为方程可以很容易地从问题中推导出来（见附录表 6）。

Variable compute only. Another intuition is that chain of thought allows the model to spend more computation (i.e., intermediate tokens) on harder problems. To isolate the effect of variable computation from chain-of-thought reasoning, we test a configuration where the model is prompted to output a only sequence of dots (. . .) equal to the number of characters in the equation needed to solve the problem. This variant performs about the same as the baseline, which suggests that variable computation by itself is not the reason for the success of chain-of-thought prompting, and that there appears to be utility from expressing intermediate steps via natural language.

Chain of thought after answer. Another potential benefit of chain-of-thought prompting could simply be that such prompts allow the model to better access relevant knowledge acquired during pretraining. Therefore, we test an alternative configuration where the chain of thought prompt is only given after the answer, isolating whether the model actually depends on the produced chain of thought to give the final answer. This variant performs about the same as the baseline, which suggests that the sequential reasoning embodied in the chain of thought is useful for reasons beyond just activating knowledge.

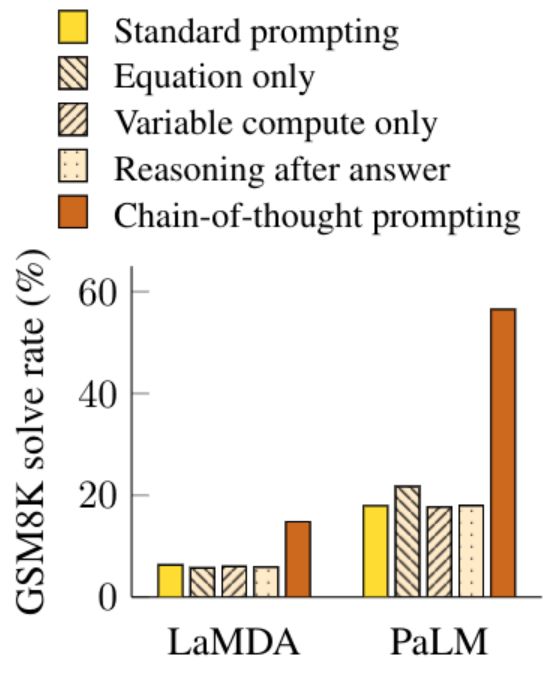


Figure 5: Ablation study for different variations of prompting using LaMDA 137B and PaLM 540B. Results for other datasets are given in Appendix Table 6 and Table 7.

3.4 Robustness of Chain of Thought

Sensitivity to exemplars is a key consideration of prompting approaches—for instance, varying the permutation of few-shot exemplars can cause the accuracy of GPT-3 on SST-2 to range from near chance (54.3%) to near state of the art (93.4%) (Zhao et al., 2021). In this final subsection, we evaluate robustness to chains of thought written by different annotators. In addition to the results above, which used chains of thought written by an Annotator A, two other co-authors of this paper (Annotators B and C) independently wrote chains of thought for the same few-shot exemplars (shown in Appendix H). Annotator A also wrote another chain of thought that was more concise than the original, following the style of solutions given in Cobbe et al. (2021).[1]

Figure 6 shows these results for LaMDA 137B on GSM8K and MAWPS (ablation results for other datasets are given in Appendix Table 6 / Table 7). Although there is variance among different chain of thought annotations, as would be expected when using exemplar-based prompting (Le Scao and Rush, 2021; Reynolds and McDonell, 2021; Zhao et al., 2021), all sets of chain of thought prompts outperform the standard baseline by a large margin. This result implies that successful use of chain of thought does not depend on a particular linguistic style.

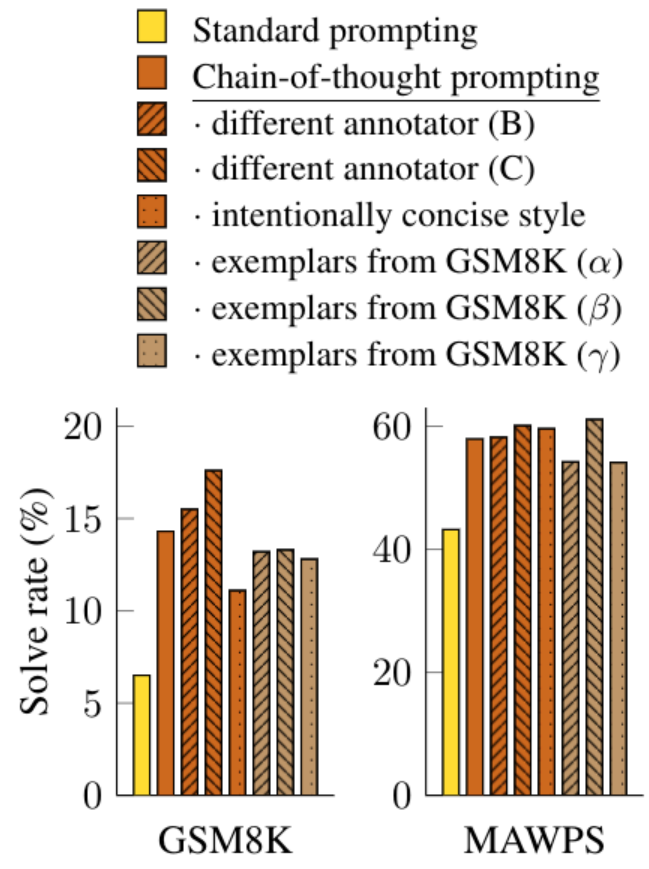


Figure 6: Chain-of-thought prompting has variance for different prompt examples (as expected) but outperforms standard prompting for various annotators as well as for different exemplars.

To confirm that successful chain-of-thought prompting works for other sets of exemplars, we also run experiments with three sets of eight exemplars randomly sampled from the GSM8K training set, an independent

[1]For instance, whereas original chain of thought uses several short sentences (*"'There were originally 9 computers. For each of 4 days, 5 more computers were added. So 5 * 4 = 20 computers were added. 9 + 20 is 29."*), the concise chain of thought would read *"5 * 4 = 20 new computers were added. So there are 9 + 20 = 29 new computers in the server room now"*.

仅限变量计算。另一种直觉是，思维链允许模型花费更多的计算（即，中间令牌）更难的问题。为了将变量计算的影响与思维链推理隔离开来，我们测试了一个配置，其中模型被提示输出一个唯一的点序列（。。。)等于解决问题所需的等式中的字符数。这个变体的表现与基线大致相同，这表明变量计算本身并不是思想链提示成功的原因，并且通过自然语言表达中间步骤似乎是有用的。

答案后的思考。思维链提示的另一个潜在好处可能只是这样的提示允许模型更好地访问在预训练期间获得的相关知识。因此，我们测试了另一种配置，其中仅在答案之后给出思维链提示，隔离模型是否实际上依赖于产生的思维链来给出最终答案。这个变体的表现与基线大致相同，这表明思维链中体现的顺序推理是有用的，其原因不仅仅是激活知识。

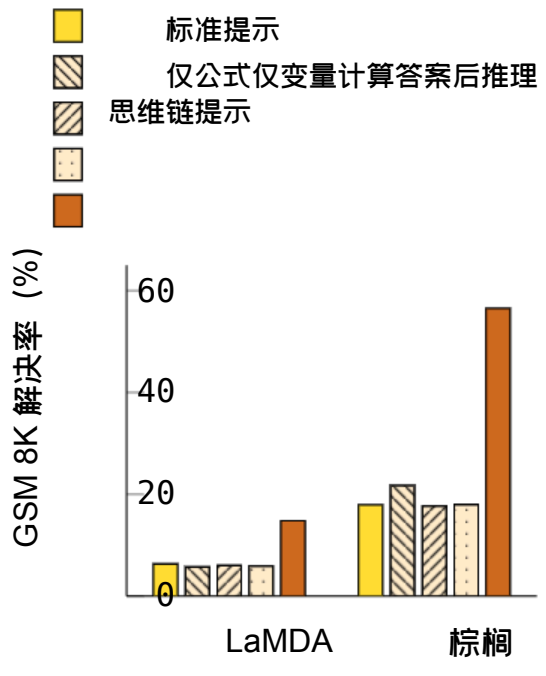
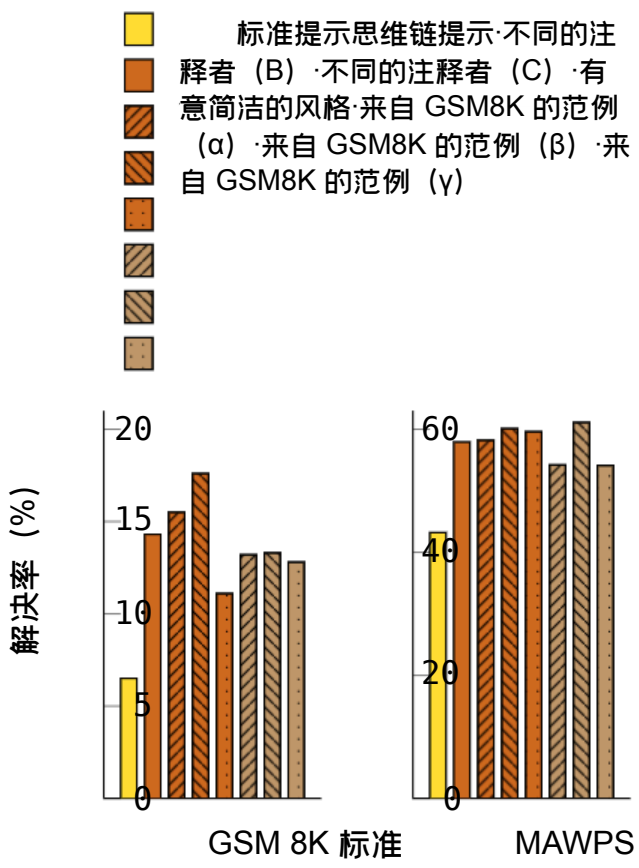


图 5：使用 LaMDA 137 B 和 PaLM 540 B 进行的不同提示变化的消融研究。其他数据集的结果见附录表 6 和表 7。

3.4 思想链的健壮性

对样本的敏感性是提示方法的关键考虑因素-例如，改变少量激发样本的排列可导致 SST-2 上 GPT-3 的准确性范围从接近偶然性（54.3%）到接近现有技术水平（93.4%）（Zhao 等人，2021 年）。在最后一小节中，我们将评估不同注释者编写的思路链的稳健性。除了上面的结果，其中使用了由注释者 A 编写的思想链，本文的另外两名合著者（注释者 B 和 C）独立地为相同的几个镜头范例编写了思想链（见附录 H）。注释者 A 还写了另一个比原来更简洁的思想链，遵循 Cobbe 等人（2021）给出的解决方案的风格。

图 6 显示了 GSM 8 K 和 MAWPS 上 LaMDA 137 B 的这些结果（其他数据集的消融结果见附录表 6 /表 7）。尽管不同的思维链注释之间存在差异，但正如使用基于范例的提示时所预期的那样（Le Scao 和 Rush，2021; Reynolds 和 McDonell，2021; Zhao 等人，2021 年），各组思路链提示的表现均大幅优于标准基线。这一结果表明，思维链的成功运用并不依赖于特定的语言风格。



图六：对于不同的提示示例，思路链提示具有差异（如预期的那样），但是对于各种注释器以及不同的示例，其优于标准提示。

为了证实成功的思维链提示对其他样本集也有效，我们还对从 GSM 8 K 训练集随机抽样的三组样本（每组八个样本）进行了实验，这是一个独立的样本集。

[1] 例如，虽然原始的思想链使用几个短句（“最初有 9 台计算机。在 4 天中，每天增加 5 台计算机。所以 5 * 4 = 20 台计算机被添加。9 + 20 等于 29”），简洁的思路是“5 * 4 = 20 台新电脑。所以现在服务器机房里有 9 + 20 = 29 台新计算机”。

source (examples in this dataset already included reasoning steps like a chain of thought).[2] Figure 6 shows that these prompts performed comparably with our manually written exemplars, also substantially outperforming standard prompting.

In addition to robustness to annotators, independently-written chains of thought, different exemplars, and various language models, we also find that chain-of-thought prompting for arithmetic reasoning is robust to different exemplar orders and varying numbers of exemplars (see Appendix A.2).

4 Commonsense Reasoning

Although chain of thought is particularly suitable for math word problems, the language-based nature of chain of thought actually makes it applicable to a broad class of commonsense reasoning problems, which involve reasoning about physical and human interactions under the presumption of general background knowledge. Commonsense reasoning is key for interacting with the world and is still beyond the reach of current natural language understanding systems (Talmor et al., 2021).

Benchmarks. We consider five datasets covering a diverse range of commonsense reasoning types. The popular **CSQA** (Talmor et al., 2019) asks commonsense questions about the world involving complex semantics that often require prior knowledge. **StrategyQA** (Geva et al., 2021) requires models to infer a multi-hop strategy to answer questions. We choose two specialized evaluation sets from the BIG-bench effort (BIG-bench collaboration, 2021): **Date** Understanding, which involves inferring a date from a given context, and **Sports** Understanding, which involves determining whether a sentence relating to sports is plausible or implausible. Finally, the **SayCan** dataset (Ahn et al., 2022) involves mapping a natural language instruction to a sequence of robot actions from a discrete set. Figure 3 shows examples with chain of thought annotations for all datasets.

Prompts. We follow the same experimental setup as the prior section. For CSQA and StrategyQA, we randomly selected examples from the training set and manually composed chains of thought for them to use as few-shot exemplars. The two BIG-bench tasks do not have training sets, so we selected the first ten examples as exemplars in the evaluation set as few-shot exemplars and report numbers on the rest of the evaluation set. For SayCan, we use six examples from the training set used in Ahn et al. (2022) and also manually composed chains of thought.

Results. Figure 7 highlights these results for PaLM (full results for LaMDA, GPT-3, and different model scales are shown in Table 4). For all tasks, scaling up model size improved the performance of standard prompting; chain-of-thought prompting led to further gains, with improvements appearing to be largest for PaLM 540B. With chain-of-thought prompting, PaLM 540B achieved strong performance relative to baselines, outperforming the prior state of the art on StrategyQA (75.6% vs 69.4%) and outperforming an unaided sports enthusiast on sports understanding (95.4% vs 84%). These results demonstrate that chain-of-thought prompting can also improve performance on tasks requiring a range of commonsense reasoning abilities (though note that gain was minimal on CSQA).

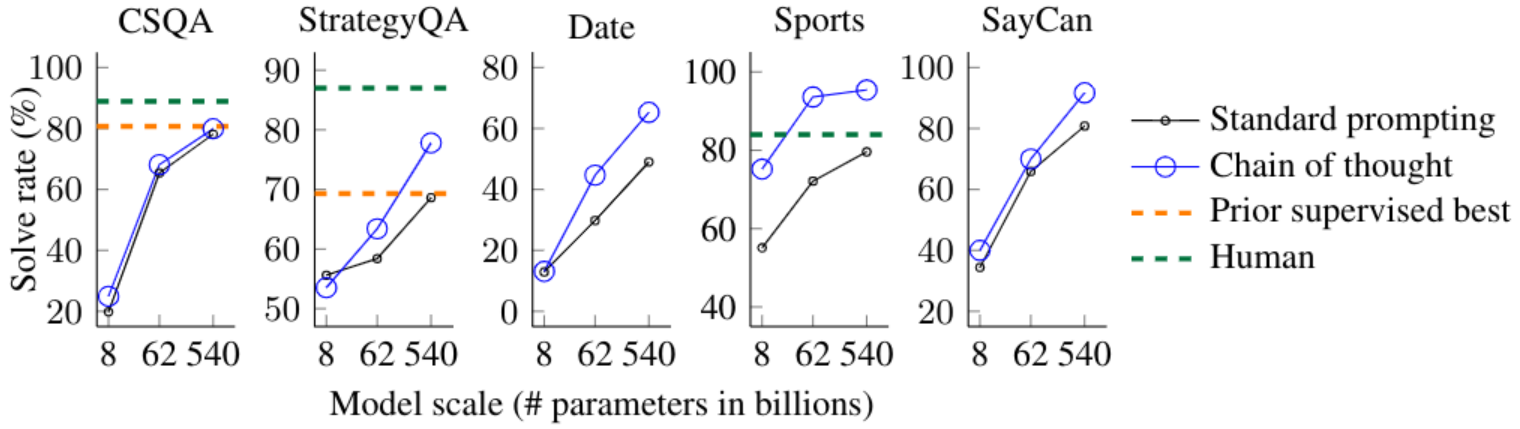


Figure 7: Chain-of-thought prompting also improves the commonsense reasoning abilities of language models. The language model shown here is PaLM. Prior best numbers are from the leaderboards of CSQA (Talmor et al., 2019) and StrategyQA (Geva et al., 2021) (single-model only, as of May 5, 2022). Additional results using various sizes of LaMDA, GPT-3, and PaLM are shown in Table 4.

[2] We sample examples ≤ 60 tokens to fit into our input context window, and also limit the examples to ≤ 2 steps to solve for a fair comparison with the eight exemplars that we composed.

图 6 显示了这些提示与我们手动编写的示例相比表现更好，也大大优于标准提示。

除了对注释者、独立编写的思维链、不同的范例和各种语言模型的鲁棒性之外，我们还发现，提示算术推理的思维链对不同的范例顺序和不同数量的范例都是鲁棒的（见附录 A.2）。

4 常识推理

尽管思维链特别适合于数学单词问题，但思维链基于语言的特性实际上使其适用于广泛的常识推理问题，这些问题涉及在一般背景知识的假设下对物理和人类相互作用的推理。常识推理是与世界交互的关键并且仍然超出了当前自然语言理解系统的范围（Talmor 等人，2021 年）。

基准。我们考虑五个数据集，涵盖了各种各样的常识推理类型。流行的 CSQA（Talmor 等人，2019 年）提出了关于世界的常识性问题，涉及复杂的语义，通常需要先验知识。策略 QA（Geva 等人，2021）要求模型推断多跳策略来回答问题。我们从 BIG-Benchage 的研究中选择了两个专门的评估集（BIG-Benchage collaboration，2021）：日期理解（Date Understanding）和体育理解（Sports Understanding），前者涉及从给定的语境中推断日期，后者涉及确定与体育相关的句子是否合理。最后，SayCan 数据集（Ahn 等人，2022）涉及将自然语言指令映射到来自离散集合的机器人动作序列。图 3 显示了所有数据集的思维链注释示例。

是的。我们遵循与上一节相同的实验设置。对于 CSQA 和 StrategyQA，我们从训练集中随机选择了一些示例，并为它们手动构建了思路链，以用作少镜头示例。两个 BIG 工作台任务没有训练集，因此我们选择前十个示例作为评估集中的样本，作为少镜头样本，并报告评估集中其余部分的编号。对于 SayCan，我们使用 Ahn 等人使用的训练集中的六个示例。

（2022），也是人工合成的思想链。

结果图 7 突出显示了 PaLM 的这些结果（LaMDA、GPT-3 和不同模型比例的完整结果见表 4）。对于所有任务，模型尺寸的扩大提高了标准提示的性能;思路链提示导致了进一步的收益，其中 PaLM 540 B 的收益似乎最大。在思路链提示下，PaLM 540 B 相对于基线实现了强劲的性能，在 StrategyQA 方面优于现有技术水平（75.6% vs 69.4%），在运动理解方面优于无辅助运动爱好者（95.4% vs 84%）。这些结果表明，在需要一系列常识性推理能力的任务中，思维链提示也可以提高绩效（尽管注意，在 CSQA 中，增益最小）。

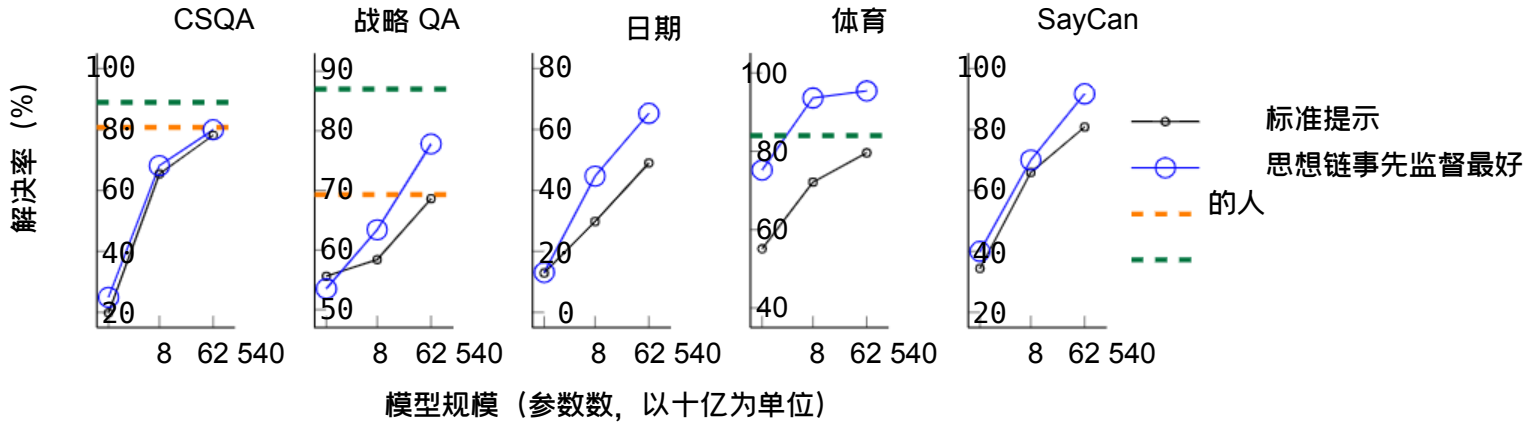


图 7：思维链提示也提高了语言模型的常识推理能力。这里显示的语言模型是 PaLM。先前的最佳数字来自 CSQA 的排行榜（Talmor 等人，2019 年）和 StrategyQA（Geva 等人，2021 年）（仅单一型号，截至 2022 年 5 月 5 日）。使用各种尺寸的 LaMDA、GPT-3 和 PaLM 的其他结果如表 4 所示。

[2] 我们对≤ 60 个 token 的示例进行采样，以适应我们的输入上下文窗口，并将示例限制为≤ 2 个步骤，以解决与我们组成的八个样本的公平比较。

5 Symbolic Reasoning

Our final experimental evaluation considers symbolic reasoning, which is simple for humans but potentially challenging for language models. We show that chain-of-thought prompting not only enables language models to perform symbolic reasoning tasks that are challenging in the standard prompting setting, but also facilitates length generalization to inference-time inputs longer than those seen in the few-shot exemplars.

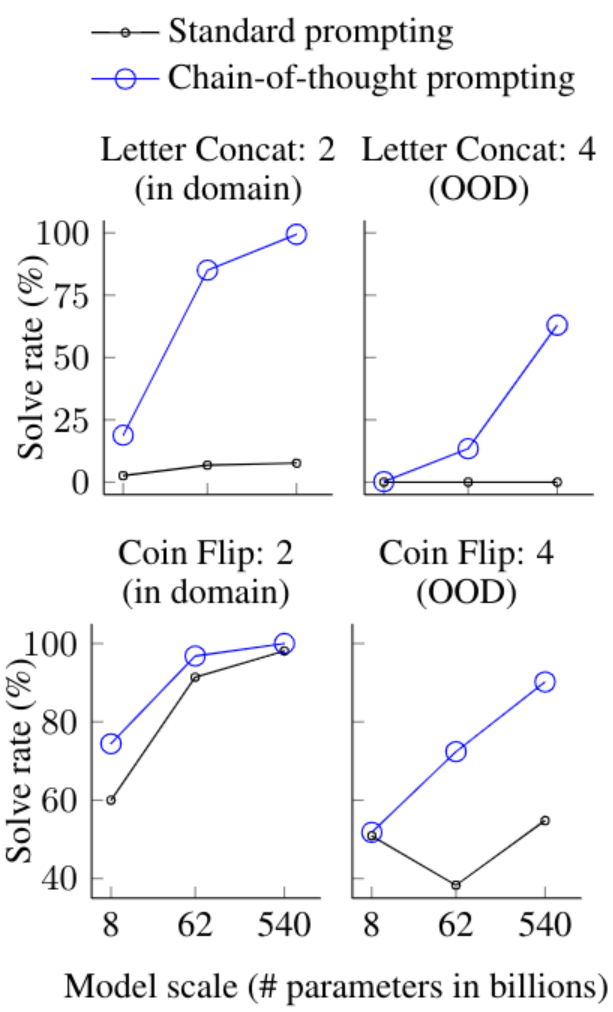


Figure 8: Using chain-of-thought prompting facilitates generalization to longer sequences in two symbolic reasoning tasks.

Tasks. We use the following two toy tasks.

- **Last letter concatenation.** This task asks the model to concatenate the last letters of words in a name (e.g., *"Amy Brown"* → *"yn"*). It is a more challenging version of first letter concatenation, which language models can already perform without chain of thought.[3] We generate full names by randomly concatenating names from the top one-thousand first and last names from name census data (`https://namecensus.com/`).
- **Coin flip.** This task asks the model to answer whether a coin is still heads up after people either flip or don't flip the coin (e.g., *"A coin is heads up. Phoebe flips the coin. Osvaldo does not flip the coin. Is the coin still heads up?"* → *"no"*).

As the construction of these symbolic reasoning tasks is well-defined, for each task we consider an *in-domain* test set for which examples had the same number of steps as the training/few-shot exemplars, as well as an *out-of-domain* (OOD) test set, for which evaluation examples had more steps than those in the exemplars. For last letter concatenation, the model only sees exemplars of names with two words, and then performs last letter concatenation on names with 3 and 4 words.[4] We do the same for the number of potential flips in the coin flip task. Our experimental setup uses the same methods and models as in the prior two sections. We again manually compose chains of thought for the few-shot exemplars for each task, which are given in Figure 3.

Results. The results of these in-domain and OOD evaluations are shown in Figure 8 for PaLM, with results for LaMDA shown in Appendix Table 5. With PaLM 540B, chain-of-thought prompting leads to almost 100% solve rates (note that standard prompting already solves coin flip with PaLM 540, though not for LaMDA 137B). Note that these in-domain evaluations are "toy tasks" in the sense that perfect solution structures are already provided by the chains of thought in the few-shot exemplars; all the model has to do is repeat the same steps with the new symbols in the test-time example. And yet, small models still fail—the ability to perform abstract manipulations on unseen symbols for these three tasks only arises at the scale of 100B model parameters.

As for the OOD evaluations, standard prompting fails for both tasks. With chain-of-thought prompting, language models achieve upward scaling curves (though performance is lower than in the in-domain setting). Hence, chain-of-thought prompting facilitates length generalization beyond seen chains of thought for language models of sufficient scale.

6 Discussion

We have explored chain-of-thought prompting as a simple mechanism for eliciting multi-step reasoning behavior in large language models. We first saw that chain-of-thought prompting improves performance by a large margin on arithmetic reasoning, yielding improvements that are much stronger than ablations and robust to different annotators, exemplars, and language models (Section 3). Next,

[3] We tested 10 common names using GPT-3 `davinci` and it got all but one correct.

[4] For names of length longer than 2 words, we concatenate multiple first and last names together.

5 符号推理

我们最后的实验评估考虑了符号推理，这对人类来说很简单，但对语言模型来说可能是一个挑战。我们发现，思维链提示不仅能使语言模型完成在标准提示设置下具有挑战性的符号推理任务，而且还能促进长度泛化到比在少镜头示例中看到的更长的推理时间输入。

任务我们使用以下两个玩具任务。

- 最后一个字母连接。此任务要求模型
  为了连接名字中单词的最后一个字母（例如，“艾米·布朗”→“yn”）。这是一个更具挑战性的首字母连接版本，语言模型已经可以在没有思想链的情况下执行。我们通过随机连接姓名普查数据（https：//namecensus.com/）中的前一千个名字和姓氏来生成全名。

- 抛硬币。此任务要求模型回答在人们抛硬币或不抛硬币后硬币是否仍然是两人对决赛（例如，“一枚硬币正朝上。菲比抛硬币。奥斯瓦尔多不会抛硬币。硬币还是两人对决赛吗？”→“否”）。

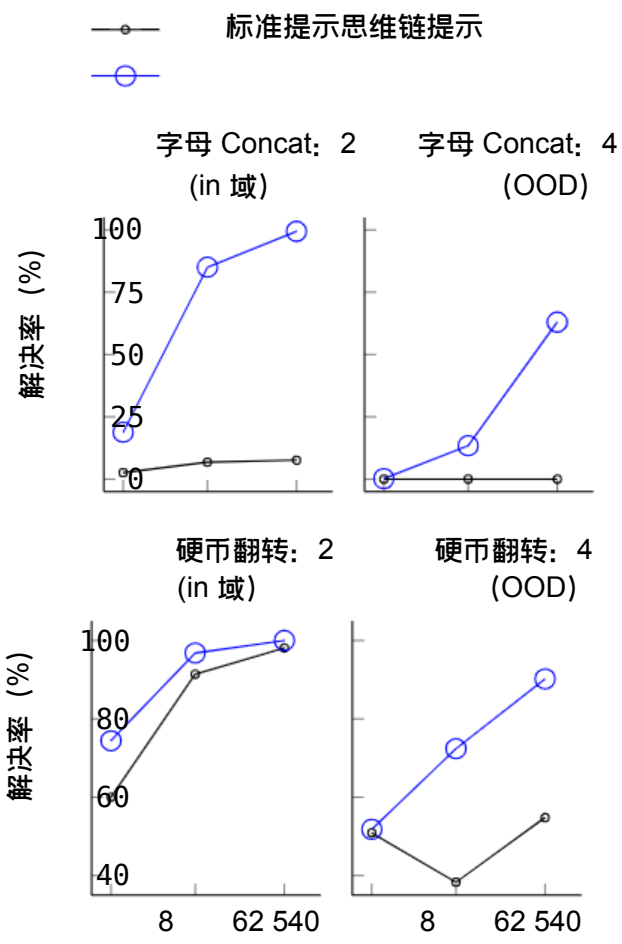


图 8：在两个符号推理任务中，使用思维链提示有助于推广到更长的序列。

由于这些符号推理任务的结构是明确定义的，因此对于每个任务，我们考虑域内测试集和域外测试集，对于域内测试集，示例与训练/少击样本具有相同数量的步骤，对于域外测试集，评估示例比样本中的步骤更多。对于最后一个字母的连接，模型只看到两个单词的名字的样本，然后对 3 个和 4 个单词的名字执行最后一个字母的连接。我们在抛硬币任务中对潜在的翻转次数做同样的事情。我们的实验设置使用了与前两节相同的方法和模型。我们再次为每项任务的少镜头示例手动构建思路链，如图 3 所示。

结果 PaLM 的这些域内和 OOD 评价结果见图 8，LaMDA 的结果见附录表 5。对于 PaLM 540 B，思路链提示导致几乎 100%的解决率（注意，标准提示已经解决了 PaLM 540 的抛硬币问题，但不适用于 LaMDA 137 B）。请注意，这些域内评估是“玩具任务”，因为在几个镜头的例子中，思维链已经提供了完美的解决方案结构;模型所要做的就是在测试时间的例子中，用新的符号重复相同的步骤。然而，小模型仍然失败--只有在 100 B 模型参数的规模下，才能为这三项任务对不可见符号执行抽象操作。

对于 OOD 评估，两项任务的标准提示均失败。通过思路链提示，语言模型实现了向上扩展曲线（尽管性能低于域内设置）。因此，对于足够大规模的语言模型，思维链提示有助于超出可见的思维链的长度概括。

6 讨论

我们已经探索了在大型语言模型中引发多步推理行为的简单机制--思想链提示。我们第一次看到，思维链提示在算术推理上大幅提高了性能，产生的改进比消融更强大，而且对不同的注释器、样本和语言模型都很健壮（第 3 节）。接下来，

[3] 我们使用 GPT-3 davinci 测试了 10 个常见名称，除了一个之外，其他都是正确的。

[4] 对于长度超过 2 个单词的名字，我们将多个名字和姓氏连接在一起。

experiments on commonsense reasoning underscored how the linguistic nature of chain-of-thought reasoning makes it generally applicable (Section 4). Finally, we showed that for symbolic reasoning, chain-of-thought prompting facilitates OOD generalization to longer sequence lengths (Section 5). In all experiments, chain-of-thought reasoning is elicited simply by prompting an off-the-shelf language model. No language models were finetuned in the process of writing this paper.

The emergence of chain-of-thought reasoning as a result of model scale has been a prevailing theme (Wei et al., 2022b). For many reasoning tasks where standard prompting has a flat scaling curve, chain-of-thought prompting leads to dramatically increasing scaling curves. Chain-of-thought prompting appears to expand the set of tasks that large language models can perform successfully—in other words, our work underscores that standard prompting only provides a lower bound on the capabilities of large language models. This observation likely raises more questions than it answers—for instance, how much more can we expect reasoning ability to improve with a further increase in model scale? What other prompting methods might expand the range of tasks that language models can solve?

As for limitations, we first qualify that although chain of thought emulates the thought processes of human reasoners, this does not answer whether the neural network is actually "reasoning," which we leave as an open question. Second, although the cost of manually augmenting exemplars with chains of thought is minimal in the few-shot setting, such annotation costs could be prohibitive for finetuning (though this could potentially be surmounted with synthetic data generation, or zero-shot generalization). Third, there is no guarantee of correct reasoning paths, which can lead to both correct and incorrect answers; improving factual generations of language models is an open direction for future work (Rashkin et al., 2021; Ye and Durrett, 2022; Wiegreffe et al., 2022, *inter alia*). Finally, the emergence of chain-of-thought reasoning only at large model scales makes it costly to serve in real-world applications; further research could explore how to induce reasoning in smaller models.

7 Related Work

This work is inspired by many research areas, which we detail in an extended related work section (Appendix C). Here we describe two directions and associated papers that are perhaps most relevant.

The first relevant direction is using intermediate steps to solve reasoning problems. Ling et al. (2017) pioneer the idea of using natural language rationales to solve math word problems through a series of intermediate steps. Their work is a remarkable contrast to the literature using formal languages to reason (Roy et al., 2015; Chiang and Chen, 2019; Amini et al., 2019; Chen et al., 2019). Cobbe et al. (2021) extend Ling et al. (2017) by creating a larger dataset and using it to finetune a pretrained language model rather than training a model from scratch. In the domain of program synthesis, Nye et al. (2021) leverage language models to predict the final outputs of Python programs via first line-to-line predicting the intermediate computational results, and show that their step-by-step prediction method performs better than directly predicting the final outputs.

Naturally, this paper also relates closely to the large body of recent work on prompting. Since the popularization of few-shot prompting as given by Brown et al. (2020), several general approaches have improved the prompting ability of models, such as automatically learning prompts (Lester et al., 2021) or giving models instructions describing a task (Wei et al., 2022a; Sanh et al., 2022; Ouyang et al., 2022). Whereas these approaches improve or augment the input part of the prompt (e.g., instructions that are prepended to inputs), our work takes the orthogonal direction of augmenting the outputs of language models with a chain of thought.

8 Conclusions

We have explored chain-of-thought prompting as a simple and broadly applicable method for enhancing reasoning in language models. Through experiments on arithmetic, symbolic, and commonsense reasoning, we find that chain-of-thought reasoning is an emergent property of model scale that allows sufficiently large language models to perform reasoning tasks that otherwise have flat scaling curves. Broadening the range of reasoning tasks that language models can perform will hopefully inspire further work on language-based approaches to reasoning.

对常识推理的实验强调了思维链推理的语言学性质如何使其普遍适用（第 4 节）。最后，我们证明了对于符号推理，思维链提示有助于将面向对象方法推广到更长的序列长度（第 5 节）。在所有的实验中，思维链推理都是通过提示现成的语言模型来引出的。在撰写本文的过程中，没有对任何语言模型进行微调。

由于模型比例的变化而出现的思维链推理已经成为一个流行的主题（Wei 等人，第 2022 条 b 款）。对于标准提示具有平坦缩放曲线的许多推理任务，思维链提示导致缩放曲线显著增加。思路链提示似乎扩展了大型语言模型可以成功执行的任务集--换句话说，我们的工作强调了标准提示只提供了大型语言模型能力的下限。这一观察结果可能提出了更多的问题，而不是回答了更多的问题，例如，随着模型规模的进一步增加，我们还能期望推理能力提高多少？还有什么其他的提示方法可以扩展语言模型可以解决的任务的范围？

至于局限性，我们首先指出，尽管思维链模仿了人类推理者的思维过程，但这并不能回答神经网络是否真的是“推理”，这是一个悬而未决的问题。其次，尽管在少数镜头设置中手动增强具有思维链的范例的成本最小，但这种注释成本可能对微调是禁止的（尽管这可能通过合成数据生成或零镜头泛化来克服）。第三，不能保证正确的推理路径，这可能导致正确和不正确的答案;改进语言模型的实际生成是未来工作的开放方向（Rashkin 等人，2021 年; Ye 和 Durrett，2022 年; Wiegreffe 等人，第 2022 号决议）。最后，思维链推理的出现仅限于大规模的模型，这使得其在现实应用中的成本很高;进一步的研究可以探索如何在较小的模型中诱导推理。

7 相关工作

这项工作受到许多研究领域的启发，我们在扩展的相关工作部分（附录 C）中详细介绍了这些研究领域。在这里，我们描述两个方向和相关的论文，也许是最相关的。

第一个相关的方向是使用中间步骤来解决推理问题。Ling et al.（2017）通过一系列中间步骤，开创了使用自然语言理据来解决数学单词问题的想法。他们的工作与使用形式语言进行推理的文献形成了鲜明的对比（Roy 等人，2015 年; Chiang 和 Chen，2019 年; Amini 等人，2019 年; Chen 等人，2019 年）。Cobbe 等人（2021 年）扩展了 Ling 等人（2017 年），创建了更大的数据集，并使用该数据集对预先训练的语言模型进行微调，而不是从头开始训练模型。在程序合成领域，Nye 等人（2021）利用语言模型，通过首先逐行预测中间计算结果来预测 Python 程序的最终输出，并表明他们的逐步预测方法比直接预测最终输出的效果更好。

当然，这篇论文也与最近关于激励的大量工作密切相关。自从 Brown 等人（2020）提出的少镜头提示的普及以来，几种通用的方法已经提高了模型的提示能力，例如自动学习提示（Lester 等人，2021）或给予描述任务的模型指令（Wei 等人，2022 a; Sanh 等人，2022;欧阳等人，2022 年）。而这些方法改进或增加了提示的输入部分（例如，在输入之前附加的指令），我们的工作采取了用思想链扩充语言模型的输出的正交方向。

8 结论

我们已经探索了思想链提示作为一种简单而广泛适用的方法，用于增强语言模型中的推理。通过对算术，符号和常识推理的实验，我们发现，思维链推理是模型规模的一个新兴属性，它允许足够大的语言模型执行推理任务，否则具有平坦的缩放曲线。扩大语言模型可以执行的推理任务的范围将有望激发基于语言的推理方法的进一步工作。

Acknowledgements

We thank Jacob Devlin, Claire Cui, Andrew Dai, and Ellie Pavlick for providing feedback on the paper. We thank Jacob Austin, Yuhuai Wu, Henryk Michalewski, Aitor Lewkowycz, Charles Sutton, and Aakanksha Chowdhery for helpful discussions. We thank Sid Maxwell for notifying us about a mistake in the manual error analysis in the original manuscript.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. 2022. Do as I can, not as I say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. 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)*, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. 2019. Giving BERT a calculator: Finding operations and arguments with reading comprehension. *EMNLP*.

Jacob Andreas, Dan Klein, and Sergey Levine. 2018. Learning with latent language. *NAACL*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

BIG-bench collaboration. 2021. Beyond the imitation game: Measuring and extrapolating the capabilities of language models. *In preparation*.

Kaj Bostrom, Xinyu Zhao, Swarat Chaudhuri, and Greg Durrett. 2021. Flexible generation of natural language deductions. *EMNLP*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *NeurIPS*.

Jonathon Cai, Richard Shin, and Dawn Song. 2017. Making neural programming architectures generalize via recursion. *ICLR*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-SNLI: Natural language inference with natural language explanations. *NeurIPS*.

Howard Chen, Jacqueline He, Karthik Narasimhan, and Danqi Chen. 2022. Can rationalization improve robustness? *NAACL*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Xinyun Chen, Chen Liang, Adams Wei Yu, Denny Zhou, Dawn Song, and Quoc V. Le. 2019. Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. *ICLR*.

Ting-Rui Chiang and Yun-Nung Chen. 2019. Semantically-aligned equation generation for solving and reasoning math word problems. 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 2656–2668, Minneapolis, Minnesota. Association for Computational Linguistics.

确认

我们感谢 Jacob Devlin、Claire Cui、Andrew Dai 和 Ellie Pavlick 对本文提供的反馈。我们感谢 Jacob Austin、Yuhuai Wu、Henryk Michalewski、Aitor Lewkowycz、Charles 萨顿和 Aakanksha Chowdhery 进行了有益的讨论。我们感谢 Sid 麦克斯韦通知我们原稿中手动错误分析的错误。

引用

Michael Ahn，Anthony Brohan，Noah Brown，Yevgen Chebotar，Omar 科尔特斯，Byron 大卫，Chelsea Finn，Keerthana Gopalakrishnan，Karol Hausman，Alex Herzog，et al. 2022.尽我所能，而不是照我说的去做：把语言根植于机器人的示能。2204. 01691. 0000. 0000. 0000. 0001. 0000. 0000. 0001. 0000. 0001. 0000.

阿依达·阿米尼、萨阿迪亚·加布里埃尔、林山川、里克·孔策尔-克齐奥尔斯基、崔业进、哈纳内·哈吉希尔齐。2019. MathQA：用操作解决可解释的数学单词问题-
基于形式主义。在计算语言学协会北美分会 2019 年会议论文集：人类语言技术，第 1 卷（长论文和短论文），明尼阿波利斯，明尼苏达州。计算语言学协会。

丹尼尔安多尔，何鲁恒，李文龙，和艾米丽皮特勒。2019.给 BERT 一个计算器：用阅读理解力查找运算和参数。EMNLP。

Jacob Andreas，Dan Klein，and Sergey Levine. 2018.用潜在语言学习。NAACL。

Jacob Austin，Augustus Odena，麦克斯韦奈，Maarten Bosma，Henryk Michalewski，大卫 Dohan，Ellen Jiang，Carrie Cai，Michael Terry，Quoc Le，et al. 2021.大型语言模型的程序合成。2108.07732.我的天啊!

大板凳合作。2021.超越模仿游戏：测量和推断语言模型的能力。准备中。

Kaj Bostrom，Xinyu Zhao，Swarat Chaudhuri，and Greg Durrett. 2021.自然语言推理的灵活生成。EMNLP。

Tom Brown，Benjamin Mann，Nick Ryder，Melanie Subbiah，Jared D Kaplan，Prafulla 达里瓦尔，Arvind Neelakantan，Pranav Shyam，Girish Sastry，阿曼达 Askell，Sandhini Agarwal，Ariel
Herbert—Voss，Gretchen Krueger，Tom Henighan，Rewon Child，Aditya Ramesh，Daniel Ziegler，Jeffrey Wu，Clemens Winter，Chris Hesse，Mark Chen，Eric Sigler，Mateusz Litwin，Scott Gray，Benjamin Chess，Jack Clark，Christopher Berner，Sam McCandlish，Alec Radford，Ilya Sutskever 和 Dario Amodei。2020.语言模型是小而精的学习者 NeurIPS。

蔡琼，理查德申，和黎明宋。2017.通过递归使神经编程架构通用化。ICLR。

Oana-Maria Camburu、Tim Rocktäschel、托马斯 Lukasiewicz 和 Phil Blunsom。2018. e-SNLI：带有自然语言解释的自然语言推理。NeurIPS。

霍华德陈，杰奎琳何，Karthik Narasimhan，和 Danqi 陈。2022.合理化能否提高稳健性？NAACL。

Mark Chen，Jerry Tworek，Heewoo Jun，Qiming Yuan，Henrique Ponde Oliveira Pinto，Jared Kaplan，Harri Edwards，Yuri Burda，Nicholas Joseph，Greg Brockman 等，2021。Evaluating large language models trained on code.（大规模的语言模型在代码上进行培训。arXiv preprint arXiv：2107.03374.（arXiv preprint)

Xinyun Chen，Chen Liang，亚当斯魏宇，Denny Zhou，Dawn Song，and Quoc V. Le. 2019.神经符号阅读器：分布式和符号表示的可扩展集成，用于阅读理解。ICLR。

蒋廷瑞和陈云农。2019.用于求解的语义对齐的方程生成

和推理数学应用题在计算语言学协会北美分会 2019 年会议论文集：人类语言技术，第 1 卷（长论文和短论文），第 2656-2668 页，明尼阿波利斯，明尼苏达州。计算语言学协会。

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. *IJCAI*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL*.

Honghua Dong, Jiayuan Mao, Tian Lin, Chong Wang, Lihong Li, and Denny Zhou. 2019. Neural logic machines. *ICLR*.

Dheeru Dua, Sameer Singh, and Matt Gardner. 2020. Benefits of intermediate annotations in reading comprehension. *ACL*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *TACL*.

Yuling Gu, Bhavana Dalvi Mishra, and Peter Clark. 2022. DREAM: Uncovering mental models behind language models. *NAACL*.

Braden Hancock, Paroma Varma, Stephanie Wang, Martin Bringmann, Percy Liang, and Christopher Ré. 2018. Training classifiers with natural language explanations. *ACL*.

Peter Hase and Mohit Bansal. 2022. When can models learn from explanations? a formal framework for understanding the roles of explanation data. *ACL*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. *EMNLP*.

Zhanming Jie, Jierui Li, and Wei Lu. 2022. Learning to reason deductively: Math word problem solving as complex relation extraction. *arXiv preprint arXiv:2203.10316*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A math word problem repository. *NAACL*.

Andrew K. Lampinen, Ishita Dasgupta, Stephanie C.Y. Chan, Kory Matthewson, Michael Henry Tessler, Antonia Creswell, James L. McClelland, Jane X. Wang, and Felix Hill. 2022. Can language models learn from explanations in context? *arXiv preprint arXiv:2204.02329*.

Yihuai Lan, Lei Wang, Qiyuan Zhang, Yunshi Lan, Bing Tian Dai, Yan Wang, Dongxiang Zhang, and Ee-Peng Lim. 2021. MWPToolkit: An open-source framework for deep learning-based math word problem solvers. *arXiv preprint arXiv:2109.00799*.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? *NAACL*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *EMNLP*.

Iddo Lev, Bill MacCartney, Christopher Manning, and Roger Levy. 2004. Solving logic puzzles: From robust processing to precise semantics. *Proceedings of the 2nd Workshop on Text Meaning and Interpretation*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *ACL*.

彼得·克拉克，奥伊文·塔夫约德，还有凯尔·理查森。2020.变形金刚是语言的软推理者。IJCAI

Karl Cobbe，Vineet Kosaraju，Mohammad Bavarian，Jacob Hilton，Reiichiro Nakano，Christopher Hesse 和 John Schulman. 2021.培训 Verifier to Solve Math Word Problems。arXiv preprint arXiv：2110.14168.（arXiv preprint)

Jacob Devlin，Ming—Wei Chang，Kenton Lee，Kristina Toutanova。2019. BERT：Pre—training of deep bidirectional transformers for language understanding，深入双向转换语言理解前的训练。NAACL 的。

Honghua Dong，Jiayuan Mao，Tian Lin，Chong Wang，Lihong Li 和 Denny Zhou。2019. Neural Logic Machines 神经逻辑机器 ICLR 的。

Dheeru Dua，Sameer Singh，和马特·加德纳. 2020.阅读理解中的中级注释的好处。ACL。

莫尔·格瓦、丹尼尔·哈沙比、埃拉德·西格尔、图沙尔·霍特、丹·罗斯和乔纳森·贝兰特。2021.做

Artistle 使用笔记本电脑吗？A question answering benchmark with implicit reasoning strategies.一个暗示推理策略的问题。Tacl。

Yuling Gu，Bhavana Dalvi Mishra，and Peter Clark（古古力先生，Bhavana Dalvi Mishra，彼得·克拉克）2022. DREAM：Uncovering mental models behind language models 揭示语言模型。NAACL 的。

Braden Hancock，Paroma Varma，Stephanie Wang，Martin Bringmann，Percy Liang 和 Christopher Ré。2018.培训经典与自然语言解释。ACL 的。

彼得·海斯和莫希特·班萨尔 2022.什么时候模型可以从解释中学习？理解解释数据的作用的正式框架。ACL。

丹·亨德里克斯、柯林·伯恩斯、索拉夫·卡达瓦斯、阿库尔·阿罗拉、史蒂文·巴萨特、埃里克·唐、道恩·宋和雅各布·斯坦哈特。2021.使用数学数据集衡量数学问题解决情况。arXiv 预印本 arXiv：2103.03874。

穆罕默德·贾瓦德·侯赛尼、汉纳内·哈吉希尔齐、奥伦·埃齐奥尼和内特·库什曼。2014.学习用动词分类解决算术单词问题。EMNLP。

解占明，李洁瑞，卢伟。2022.学习演绎推理：数学单词问题解决作为复杂关系提取。arXiv 预印本 arXiv：2203.10316。

杰瑞德·卡普兰、山姆·麦克坎迪利什、汤姆·亨尼根、汤姆·B 布朗、本杰明·切斯、雷温·蔡尔德、斯科特·格雷、亚历克·拉德福、杰弗里·吴和达里奥·阿莫代。2020.神经语言模型的标度律。arXiv 预印本 arXiv：2001.08361。

Rik Koncel-Kedziorski、Subhro Roy、Aida Amini、Nate Kushman 和 Hannaneh Hajishirzi。2016.

MAWPS：一个数学应用题库。NAACL。

安德鲁·K 放大图片作者：Stephanie C.Y.放大图片作者：Michael 亨利特斯勒，安东尼亚克雷斯韦尔，詹姆斯 L.作者声明：Jane X.王和菲利克斯·希尔 2022.语言模型可以从上下文中的解释中学习吗？arXiv 预印本 arXiv：2204.02329。

Yihuai Lan，Lei Wang，Qiyuan Zhang，Yunshi Lan，Bing Tian Dai，Yan Wang，Dongxiang Zhang，and Ee-Peng Lim. 2021. MWPToolkit：一个开源框架，用于基于深度学习的数学单词问题求解器。arXiv 预印本 arXiv：2109.00799。

特文·勒·斯科和亚历山大·拉什。2021.一个提示值多少数据点？NAACL。

布莱恩·莱斯特，拉米·艾尔-瑞福，诺亚·康斯坦。2021.缩放功能可实现高效的参数即时调整。EMNLP。

Iddo Lev，Bill MacCartney，Christopher Manning，and Roger Levy. 2004.解决逻辑难题：从强大的处理到精确的语义。第二届文本意义与解释研讨会论文集。

丽莎李翔和珀西梁。2021.前缀调优：优化生成的连续提示。ACL。

Zhengzhong Liang, Steven Bethard, and Mihai Surdeanu. 2021. Explainable multi-hop verbal reasoning through internal monologue. *NAACL*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *ACL*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Bodhisattwa Prasad Majumder, Oana-Maria Camburu, Thomas Lukasiewicz, and Julian McAuley. 2021. Rationale-inspired natural language explanations with commonsense. *arXiv preprint arXiv:2106.13876*.

Ana Marasović, Iz Beltagy, Doug Downey, and Matthew E Peters. 2022. Few-shot self-rationalization with natural language prompts. *NAACL Findings*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *ACL*.

Shen Yun Miao, Chao Chun Liang, and Keh Yih Su. 2020. A diverse corpus for evaluating and developing English math word problem solvers. *ACL*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. WT5?! Training text-to-text models to explain their predictions. *arXiv preprint arXiv:2004.14546*.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? *NAACL*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *NAACL*.

Xinyu Pi, Qian Liu, Bei Chen, Morteza Ziyadi, Zeqi Lin, Yan Gao, Qiang Fu, Jian-Guang Lou, and Weizhu Chen. 2022. Reasoning like program executors. *arXiv preprint arXiv:2201.11473*.

Piotr Piękos, Mateusz Malinowski, and Henryk Michalewski. 2021. Measuring and improving BERT's mathematical abilities by predicting the order of reasoning. *ACL*.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training Gopher. *arXiv preprint arXiv:2112.11446*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Dheeraj Rajagopal, Vidhisha Balachandran, Eduard H. Hovy, and Yulia Tsvetkov. 2021. SelfExplain: A self-explaining architecture for neural text classifiers. *EMNLP*.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! Leveraging language models for commonsense reasoning. *ACL*.

Zhengzhong Liang，Steven Bethard，and Mihai Surdeanu. 2021.通过内部独白进行可解释的多跳言语推理。NAACL。

王玲，Dani Yogatama，Chris Dyer 和 Phil Blunsom。2017.程序归纳的基本原理生成：学习解决和解释代数字的问题。ACL。

刘鹏飞、袁伟哲、傅金兰、姜正宝、林博明、格雷厄姆·诺伊比格。2021.
预训练、提示和预测：自然语言处理中提示方法的系统综述。arXiv 预印本 arXiv：2107.13586。

Bodhisattwa Prasad Majumder，Oana-Maria Camburu，托马斯 Lukasiewicz 和 Julian McAuley。
2021. 以常识为灵感的自然语言解释。arXiv 预印本 arXiv：2106.13876。

安娜·马拉索维奇，伊兹·贝尔塔吉，道格·唐尼和马修·E·彼得斯。2022.用自然语言提示进行少量的自我合理化。NAACL 调查结果。

约书亚·梅内兹，沙希·纳拉扬，贝恩德·博内特和瑞安·麦克唐纳。2020.论抽象概括的忠实性与真实性。在 ACL。

沈云淼、赵春良、柯依素。2020.一个评估和开发英语数学应用题解决者的多样化语料库。ACL。

Sewon Min、Xinxi Lyu、Ari Holtzman、Mikel Artetxe、Mike 刘易斯、Hannaneh Hajishirzi 和 Luke Zettlemoyer。2022.反思示范的作用：是什么让情境学习发挥作用？

arXiv 预印本 arXiv：2202.12837。

沙兰·纳朗、科林·拉菲尔、凯瑟琳·李、亚当·罗伯茨、诺亚·菲德尔和卡里什马·马尔坎。
2020. WT5？训练文本到文本模型来解释他们的预测。arXiv 预印本 arXiv：2004.14546。

麦克斯韦奈、安德斯·约翰·安德雷森、盖伊·古尔-阿里、亨里克·米卡莱夫斯基、雅各布·奥斯汀、大卫
比伯，大卫多汗，Aitor Lewkowycz，Maarten Bosma，大卫卢安，等 2021。展示你的作品：使用语言模型进行中间计算的 Sockchpad。arXiv 预印本 arXiv：2112.00114。

欧阳龙，吴建杰，徐江，迪奥戈阿尔梅达，卡罗尔 L.温赖特，帕梅拉·米什金，张冲，桑希尼·阿加瓦尔，卡塔琳娜·斯拉马，亚历克斯·雷，等 2022.训练语言模型遵循指令并提供人类反馈。arXiv 预印本 arXiv：2203.02155。

阿图尔·帕特尔、萨特维克·巴塔米什拉和纳文·戈亚尔。2021. NLP 模型真的能够解决简单的数学单词问题吗？NAACL。

马修·E Peters，Mark Neumann，Mohit Iyyer，Matt Gardner，Christopher Clark，Wendon Lee，and Luke Zettlemoyer. 2018.深层语境化的词语表征。NAACL。

Xinyu Pi，Qian Liu，Bei Chen，Morteza Ziyadi，Zeqi Lin，Yan Gao，Qiang Fu，Jian-Guang Lou，and Weizhu Chen. 2022.像程序执行者一样推理。arXiv 预印本 arXiv：2201.11473。

Piotr Pi Pizzekos，Mateusz Malinowski，and Henryk Michalewski. 2021.通过预测推理顺序来衡量和提高 BERT 的数学能力。ACL。

杰克·W Rae，塞巴斯蒂安 Borgeaud，特雷弗蔡，凯蒂 Millican，乔丹霍夫曼，弗朗西斯宋，约翰

Aslanides，Sarah 亨德森，Roman Ring，Susannah Young，et al. 2021. Scaling language models：Methods，analysis & insights from training Gopher. arXiv 预印本 arXiv：2112.11446。

Colin Raffel、Noam Shazeer、Adam Roberts、凯瑟琳李、沙兰纳朗、Michael Matena、Yanqi Zhou、Wei Li 和 Peter J Liu。2020.使用统一的文本到文本 Transformer 探索迁移学习的局限性。机器学习研究杂志，21：1-67。

放大图片作者：Dheeraj Rajagopal，Vidhisha Balachandran，Eduard H. Hovy 和 Yulia Tsvetkov。2021. SelfExplain：神经文本分类器的自我解释架构。EMNLP。

Nazneen Fatema Rajani，Bryan McCann，Caiming Xiong，and Richard Socher. 2019.解释一下！利用语言模型进行常识推理。ACL。

Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. NumNet: Machine reading comprehension with numerical reasoning. *EMNLP*.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2021. Measuring attribution in natural language generation models. *arXiv preprint arXiv:2112.12870*.

Gabriel Recchia. 2021. Teaching autoregressive language models complex tasks by demonstration. *arXiv preprint arXiv:2109.02102*.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2022. A recipe for arbitrary text style transfer with large language models. *ACL*.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. *EMNLP*.

Subhro Roy, Tim Vieira, and Dan Roth. 2015. Reasoning about Quantities in Natural Language. *TACL*.

Mohammed Saeed, Naser Ahmadi, Preslav Nakov, and Paolo Papotti. 2021. RuleBERT: Teaching soft rules to pre-trained language models. *EMNLP*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask prompted training enables zero-shot task generalization. *ICLR*.

Jianhao Shen, Yichun Yin, Lin Li, Lifeng Shang, Xin Jiang, Ming Zhang, and Qun Liu. 2021. Generate & rank: A multi-task framework for math word problems. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. *NAACL*.

Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. *NeurIPS*.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2021. CommonsenseQA 2.0: Exposing the limits of ai through gamification. *NeurIPS Track on Datasets and Benchmarks*.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. 2022. Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. LaMDA: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022a. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022b. Benchmarking generalization via in-context instructions on 1,600+ language tasks. *arXiv preprint arXiv:2204.07705*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. *ICLR*.

Qiu Ran，Yankai Lin，Peng Li，Jie Zhou，and Zhiyuan Liu. 2019. NumNet：具有数字推理的机器阅读理解。EMNLP。

Hannah Rashkin、Vitaly Nikolaev、Matthew Lamm、Michael 柯林斯、Dipanjan Das、Slav Petrov、Gaurav Singh 托马尔、Iulia Turc 和大卫 Reitter。2021.在自然语言生成模型中测量归因。arXiv 预印本 arXiv：2112.12870。

加布里埃尔·雷基亚 2021.自回归语言教学模型演示复杂的任务。

arXiv 预印本 arXiv：2109.02102。

艾米丽赖夫，达芙妮伊波利托，安袁，安迪科南，克里斯卡利森伯奇，和贾森魏。2022.
使用大型语言模型进行任意文本样式传输的秘诀。ACL。

拉莉亚·雷诺兹和凯尔·麦克唐纳。2021.大型语言模型的快速编程：超越
少拍模式 2021 年 CHI 计算机系统人为因素会议的扩展摘要。

苏布洛·罗伊和丹·罗斯 2015.解决一般算术题。EMNLP。

苏布洛·罗伊，蒂姆·维埃拉，丹·罗斯。2015.自然语言中的推理。
TACL。

Mohammed Saeed，Naser Ahmadi，Preslav Nakov，and Paolo Papotti. 2021. RuleBERT：向预先训练的语言模型教授软规则。EMNLP。

维克托·桑（Victor Sanh）、阿尔伯特·韦伯森（Albert Webson）、科林·拉菲尔（Colin Raffel）、斯蒂芬·H. Bach，Lintang Sutawika，Zaid Alyafeai，Antoine Chaffin，Arnaud Stiegler，Teven Le Scao，Arun Raja，et al. 2022.多任务提示训练可实现零触发任务概括。ICLR。

Jianhao Shen，Yichun Yin，Lin Li，Lifeng Shang，Xin Jiang，Ming Zhang，and Qun Liu. 2021.

Generate & rank：一个数学应用题的多任务框架。计算语言学协会（Association for Computational Linguistics：EMNLP 2021）

Alon Talmor，Jonathan Herzig，Nicholas Lourie，and Jonathan Berant. 2019. CommonsenseQA：针对常识知识的问答挑战。NAACL。

Alon Talmor，Oyvind Tafjord，Peter Clark，Yoav Goldberg，and Jonathan Berant. 2020. Leap-ofthought：教预先训练的模型系统地推理隐含的知识。NeurIPS。

Alon Talmor，Ori Yoran，罗南勒布拉斯，钱德拉 Bhagavatula，Yoav Goldberg，Yejin Choi，和 Jonathan Berant. 2021.常识问答 2.0：通过游戏化暴露人工智能的极限。

NeurIPS 跟踪数据集和基准测试。

Yi Tay，Mostafa Dehghani，Vinh Q Tran，Xavier Garcia，达拉 Bahri，Tal Schuster，Huaixiu Steven Zheng，Neil Houlsby，and Donald Metzler. 2022.统一语言学习范式。arXiv 预印本 arXiv：2205.05131。

Romal Thoppilan，丹尼丹尼尔德 Freitas，Jamie Hall，Noam Shazeer，Apoorv Kulshreshtha，Heng-Tze Cheng，Alicia Jin，Taylor Bos，Leslie Baker，Yu Du，et al. 2022. LaMDA：对话应用程序的语言模型。arXiv 预印本 arXiv：2201.08239。

Xuechi Wang，Jason Wei，Dale Schuurmans，Quoc Le，艾德迟，和 Denny Zhou. 2022 年 a。
自我一致性改进了语言模型中的思维推理链。arXiv 预印本 arXiv：2203.11171。

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022b.
通过 1，600 多个语言任务的上下文指令进行基准泛化。arXiv 预印本 arXiv：2204.07705。

放大图片作者：Jason Wei，Maarten Bosma，Vincent Y.作者：赵，Kelvin Guu，亚当斯于伟，Brian Lester，杜楠，Andrew M. Dai 和 Quoc V. Le 的著作。2022 年 a。经过微调的语言模型是零机会学习者。ICLR。

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022b. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Sarah Wiegreffe, Jack Hessel, Swabha Swayamdipta, Mark Riedl, and Yejin Choi. 2022. Reframing human-AI collaboration for generating free-text explanations. *NAACL*.

Sarah Wiegreffe and Ana Marasović. 2021. Teach me to explain: A review of datasets for explainable NLP. *NeurIPS*.

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2021. Measuring association between labels and free-text rationales. *EMNLP*.

Tongshuang Wu, Ellen Jiang, Aaron Donsbach, Jeff Gray, Alejandra Molina, Michael Terry, and Carrie J Cai. 2022a. PromptChainer: Chaining large language model prompts through visual programming. *CHI Extended Abstracts*.

Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022b. AI chains: Transparent and controllable human-AI interaction by chaining large language model prompts. *CHI*.

Yujun Yan, Kevin Swersky, Danai Koutra, Parthasarathy Ranganathan, and Milad Hashemi. 2020. Neural execution engines: Learning to execute subroutines. *NeurIPS*.

Huihan Yao, Ying Chen, Qinyuan Ye, Xisen Jin, and Xiang Ren. 2021. Refining language models with compositional explanations. *NeurIPS*.

Xi Ye and Greg Durrett. 2022. The unreliability of explanations in few-shot in-context learning. *arXiv preprint arXiv:2205.03401*.

Yordan Yordanov, Vid Kocijan, Thomas Lukasiewicz, and Oana-Maria Camburu. 2021. Few-shot out-of-domain transfer learning of natural language explanations. *arXiv preprint arXiv:2112.06204*.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. *NAACL*.

Wojciech Zaremba and Ilya Sutskever. 2014. Learning to execute. *arXiv preprint arXiv:1410.4615*.

Eric Zelikman, Yuhuai Wu, and Noah D. Goodman. 2022. STaR: Bootstrapping reasoning with reasoning. *arXiv preprint arXiv:2203.14465*.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *ICML*.

Wangchunshu Zhou, Jinyi Hu, Hanlin Zhang, Xiaodan Liang, Maosong Sun, Chenyan Xiong, and Jian Tang. 2020. Towards interpretable natural language understanding with explanations as latent variables. *NeurIPS*.

Jason Wei、Yi Tay、Rishi Bommasani、Colin Raffel、Barret Zoph、塞巴斯蒂安·博吉奥、Dani Yogatama、Maarten Bosma、Denny Zhou、Donald Metzler 等人。2022 b。大型语言模型的涌现能力。机器学习研究论文集。

Sarah Wiegreffe，Jack Hessel，Swabha Swayamdipta，Mark 里德尔，and Yejin Choi. 2022.为生成自由文本解释而重新构建人类-AI 协作。NAACL。

莎拉·维格里夫和安娜·马拉索维。2021.教我解释：可解释的自然语言处理的数据集回顾。NeurIPS。

莎拉·维格里夫，安娜·马拉索维奇，诺亚·A。史密斯 2021.测量标签和自由文本原理之间的关联。EMNLP。

Tongshuang Wu，Ellen Jiang，Aaron 东斯巴赫，Jeff Gray，Alejandra Molina，Michael Terry，and Carrie J Cai. 2022 年 a.提示链：通过可视化编程链接大型语言模型提示。CHI Extended Abstracts.

Tongshuang Wu，Michael Terry，and Carrie Jun Cai. 2022 b.第二次定期报告 AI 链：通过链接大型语言模型提示，实现透明和可控的人机交互。CHI.

Yujun Yan，Kevin Swersky，Danai Koutra，Parthasarathy Ranganathan，and Milad Hashemi. 2020.
神经执行引擎：学习执行子程序。NeurIPS。

姚惠涵、陈英、叶钦元、金锡森、项仁。2021.用组合解释精炼语言模型。NeurIPS。

席叶和格雷格·德雷特。2022.少数情境学习中解释的不可靠性。
arXiv 预印本 arXiv：2205.03401。

Yordan Yordanov，Vid Kocijan，托马斯 Lukasiewicz 和 Oana-Maria Camburu。2021.少样本
自然语言解释的域外迁移学习。arXiv 预印本 arXiv：2112.06204。

奥马尔·扎伊丹杰森·纳和克莉丝汀·皮亚特科。2007.使用“注释器原理”改进文本分类的机器学习。NAACL。

Wojciech Zaremba 和 Ilya Sutskever。2014.学习执行。Arxiv 预印本 Arxiv：1410.4615.

Eric Zelikman，Yuhuai Wu 和 Noah D. Goodman. 2022.星星：Bootstrapping reasoning with reasoning。2203.14465.我的天啊!

托尼·Z 赵耀、埃里克·华莱士、石峰、丹·克莱恩、萨米尔·辛格。2021.用途：提高语言模型的少数镜头性能。ICML。

Wangchunshu Zhou，Jinyi Hu，Hanlin Zhang，Xiaodan Liang，Maosong Sun，Chenyan Xiong，and Jian Tang. 2020.以解释为潜在变量的可解释自然语言理解。NeurIPS。