大语言模型综述

赵鑫，周昆 *，李军毅 *，唐天一，王晓磊，侯宇蓬，闵映乾，张北辰，张君杰，董梓灿，都一凡，杨晨，陈昱硕，陈志朋，蒋锦昊，任瑞阳，李依凡，汤昕宇，刘子康，刘沛羽，聂建云，文继荣

摘要—自从 20 世纪 50 年代图灵测试被提出以来，人类一直在探索如何用机器掌握语言智能。语言本质上是一种由语法规则支配的复杂的人类表达系统，开发有能力理解和掌握一门语言的人工智能 (AI) 算法是一个重大挑战。作为一种主要的语言理解和生成方法，语言建模在过去的二十年中得到了广泛的研究，其从统计语言模型发展为神经语言模型。近年来，通过在大规模语料库上预训练，基于 Transformer 架构的预训练语言模型在解决各种自然语言处理任务方面表现出强大的能力。由于研究人员发现扩大模型规模可以提高模型能力，因此他们通过将参数增加到更大的尺寸来进一步研究该效应。有趣的是，当参数规模超过一定水平时，这些规模扩大的语言模型的性能不仅得到了显著提升，而且还表现出一些小规模语言模型（如 BERT）所不具备的特殊能力（如上下文学习）。为了区分不同参数规模下的语言模型，研究团体创造了术语——大语言模型（LLM）代指大型的预训练语言模型（如包含数百亿或数千亿个参数）。近年来，学术界和业界极大的推进了针对大语言模型的研究，并在该方向取得了显著的进展，如 ChatGPT（一种基于 LLM 开发的强大 AI 聊天机器人）的推出，引起了社会的广泛关注。大语言模型的技术发展对整个 AI 社区产生了重要影响，这将彻底改变我们开发和使用 AI 算法的方式。考虑到这一快速的技术进步，在本篇综述中，我们通过介绍大语言模型的背景、主要发现和主流技术来回顾近年来的进展。我们特别关注大语言模型的四个主要方面，即预训练、适配微调、应用和能力评估。此外，我们还总结了开发大语言模型的可用资源，并讨论了未来可行的发展方向。本文提供了关于大语言模型的最新文献综述，期望能为研究人员和工程师提供帮助。

Index Terms—大语言模型，涌现能力，适配微调，应用，对齐，能力评估

1 引言

语言是人类表达和交流的突出能力，它在儿童早期发展并在一生中不断演变 [1, 2]。然而，机器不能自然地掌握以人类语言形式理解和交流的能力，除非配备了强大的人工智能算法。实现这一目标，让机器像人类一样阅读、写作和交流，一直是一个长期的研究挑战 [3]。

从技术上讲，语言建模是提高机器语言智能的主要方法之一。一般来说，语言建模旨在对词序列的生成概率进行建模，以预测未来（或缺失）单词的概率。语言建模的研究在文献中受到了广泛关注，可以分为四个主要发展阶段：

• 统计语言模型 *(SLM)*：SLMs [4–7] 基于统计学习方法开发，并在 20 世纪 90 年代兴起。其基本思想是基于马尔可夫假设建立词预测模型，例如根据最近的上下文预测下一个词。具有固定上下文长度 n 的 SLM 也称为 n-gram 语言模型，例如 bigram 和 trigram 语言模型。SLM 已被广泛应用于提高信息检索 [8, 9] 和自然语言处理 [10–12] 的任务性能。然而，它们通常受到维数灾难的困扰：由于需要估计指数级数量的转换概率，因此很难准确估计高阶语言模型。因此，专门设计的平滑策略，如回退估计 [13] 和 Good–Turing 估计 [14] 已被引入以缓解数据稀疏问题。

• 神经语言模型 *(NLM)*：NLMs [15–17] 使用神经网络（例如循环神经网络）来刻画词序列的概率。作为一个显著贡献，[15] 的工作引入了词的分布式表示概念，并在聚合上下文特征（即分布式词向量）的条件下构建词预测函数。通过扩展学习词或句子有效特征的想法，已有研究开发了一种通用神经网络方法来为各种自然语言处理任务构建统一解决方案 [18]。此外，word2vec [19, 20] 被提出来构建一个简化的浅层神经网络，用于学习分布式词表示，这些表示在各种自然语言处理任务中被证明非常有效。这些研究开创了将语言模型用于表示学习（超越词序列建模），对自然语言处理领域产生了重要影响。

• 预训练语言模型 *(PLM)*：作为早期尝试，ELMo [21] 被提出来通过预训练一个双向 LSTM（biLSTM）网络（而不是学习固定的词表示）来捕捉上下文感知的词表示，然后根据特定的下游任务微调 biLSTM 网络。进一步，基于自注意力机制的高度并行化 Transformer 架构 [22]，BERT [23] 作为双向语言模型，在大规模无标签语料库上使用专门设计的预训练任务。这些预训练的上下文感知词表示作为通用语义特征非常有效，极大地提高了自然语言处理任务的性能。这项研究激发了大量后续工作，确立了“预训练和微调”学习范式。遵循这一范式，已经建立了大量关于预训练语言模型的研究，引入了不同的架构 [24, 25]（例如 GPT-2 [26] 和 BART [24]），或者改进的预训练策略 [27–29]。在这个范式中，通常需要对

- *GitHub 链接: https://github.com/RUCAIBox/LLMSurvey*
- 英文原文链接：*https://arxiv.org/abs/2303.18223*
- 注：本文为英文综述论文《*A Survey of Large Language Models*》的翻译稿件（版本 *v4*，非最新版本）。本中文版本系使用“大模型翻译 + 少量人工复核”完成，仅用来方便读者对应参考，暂不用做投稿以及其他用途。由于时间所限，更为准确、完善的翻译工作还在规划中。请读者们以英文文章为主进行阅读，本文仅供参考，并不进行实时更新和维护，不保证与英文版本一一对应关系。**本文未经许可，不得以任何形式进行转发，或者拷贝使用相关内容。**

预训练语言模型进行微调以适应不同的下游任务。

• 大语言模型 *(LLM)*: 研究人员发现，扩展预训练语言模型（例如扩展模型大小或数据大小）通常会提高下游任务的模型容量（即遵循扩展定律 [30]）。许多研究通过训练越来越大的 PLM（例如 175B 参数的 GPT-3 和 540B 参数的 PaLM）来探索性能极限。尽管扩展主要在模型大小方面进行（具有类似的架构和预训练任务），但这些大尺寸的预训练语言模型表现出与较小的预训练语言模型（如 330M 参数的 BERT 和 1.5B 参数的 GPT-2）不同的行为，并在解决一系列复杂任务中展示了惊人的能力（称为涌现能力）。例如，GPT-3 可以通过上下文学习解决少样本任务，而 GPT-2 则表现不佳。因此，研究界为这些大型预训练语言模型命名为“大语言模型 *(LLM)*”[1] [31–34]。LLM 的一个显著应用是 *ChatGPT*[2]，它将 GPT 系列的 LLM 应用于对话，展现了惊人的与人类对话的能力。

在现有文献中，PLM 已经得到了广泛的讨论和调研 [35–38]，而 LLM 很少以系统的方式进行回顾。为了激发我们的调研，我们首先强调 LLM 和 PLM 之间的三个主要区别。首先，LLM 表现出一些令人惊讶的涌现能力，这些能力可能在以前较小的 PLM 中没有观察到。这些能力是语言模型在复杂任务上表现的关键，使得人工智能算法具有前所未有的强大和有效性。其次，LLM 将彻底改变人类开发和使用人工智能算法的方式。与小型 PLM 不同，访问 LLM 的主要方法是通过提示接口（例如 GPT-4 API）。人们必须了解 LLM 的工作原理，并以 LLM 能够遵循的方式形式化他们的任务。第三，LLM 的发展不再明确区分研究和工程。训练 LLM 需要在大规模数据处理和分布式并行训练方面具有丰富的实践经验。为了开发出有能力的 LLM，研究人员必须解决复杂的工程问题，与工程师合作或成为工程师。

如今，LLM 对 AI 社区产生了重大影响，ChatGPT 和 GPT-4 的出现促使人们重新思考通用人工智能（AGI）的可能性。OpenAI 已经发布了一篇名为“*Planning for AGI and beyond*”的技术文章，讨论了实现 AGI 的短期和长期计划 [39]，而一篇更近期的论文认为 GPT-4 可能被视为 AGI 系统的早期版本 [40]。AI 研究领域正因 LLM 的迅速发展而发生革命性变革。在自然语言处理领域，LLM 可以在一定程度上作为通用语言任务解决器，其研究范式已经转向使用 LLM。在信息检索领域，传统搜索引擎正受到通过 AI 聊天机器人（即 ChatGPT）搜索新信息的挑战，而 *New Bing*[3]展示了一个初步的基于 LLM 增强搜索结果的研究尝试。在计算机视觉领域，研究人员试图开发类似 ChatGPT 的视觉-语言模型，以更好地为多模态对话提供服务 [41–44]，GPT-4 [45] 已经通过整合视觉信息支持多模态输入。这一新技术浪潮可能会带来一个基于 LLM 的繁荣实际应用生态系统。例如，Microsoft 365 正在被 LLM（如 Copilot）赋能以自动化办公工作，而 OpenAI 支持在 ChatGPT 中使用插件来实现特殊功能。

尽管取得了进步和影响，但 LLM 的基本原理尚未得到充分探索。首先，为什么涌现能力会出现在 LLM 中，而不是较小的 PLM 中，这仍然是个谜。作为一个更普遍的问题，缺乏对 LLM 优越能力的关键因素进行深入、详细调查的研究。研究 LLM 何时以及如何获得这些能力非常重要 [46]。尽管对这个问题已有一些有意义的讨论 [46, 47]，但仍需要更多原则性的研究来揭示 LLM 的“秘密”。其次，研究界很难训练出有能力的 LLM。由于计算资源的巨大需求，为了研究各种策略对训练 LLM 的效果，进行重复、消融研究的成本非常高。实际上，LLM 主要由工业界训练，许多重要的训练细节（如数据收集和清理）并未向公众透露。第三，将 LLM 与人类价值观或偏好对齐是具有挑战性的。尽管有能力，LLM 也可能产生有毒的、虚构的或有害的内容。消除使用 LLM 的潜在风险需要有效且高效的控制方法 [45]。

面对机遇和挑战，我们需要更多关注 LLM 的研究和发展。为了提供对 LLM 的基本了解，本综述从四个主要方面对 LLM 的最近进展进行文献综述，包括预训练（如何预训练出一个有能力的 LLM）、适应微调（如何从有效性和安全性两个角度有效地微调预训练的 LLM）、使用（如何利用 LLM 解决各种下游任务）以及能力评估（如何评估 LLM 的能力和现有的经验性发现）。我们彻底梳理了文献，总结了 LLM 的关键发现、技术和方法。对于这篇综述，我们还创建了一个 GitHub 项目网站，收集了关于 LLM 的支持资源，链接为 https://github.com/RUCAIBox/LLMSurvey. 我们也了解到了一些关于 PLM 或 LLM 的相关综述文章 [31, 35, 37, 38, 42, 48–54]。这些论文要么讨论 PLM，要么讨论 LLM 的某些特定（或通用）方面。与它们相比，我们关注开发和使用 LLM 的技术和方法，并为 LLM 的重要方面提供相对全面的参考。

本综述的其余部分安排如下：第 2 节介绍 LLM 的背景，包括术语、设置、资源和大纲概述，接着在第 3 节总结开发 LLM 的可用资源。第 4、5、6 和 7 节分别从预训练、适应微调、使用和能力评估四个方面回顾和总结了最近的进展。最后，在第 8 节中，我们通过总结主要发现并讨论未来工作的剩余问题来结束这次综述。

2 概述

在本节中，我们介绍了 LLM 的背景，包括其关键术语、能力和技术。

背景： 通常，大语言模型 (LLM) 指包含数百亿（或更多）参数的语言模型[4]，这些模型在大量的文本数据 [31] 上进行训练，

1. 请注意，LLM 并不一定比小型 PLM 更有能力，而且涌现能力在某些 LLM 中可能不会出现。

2. https://openai.com/blog/chatgpt/

3. https://www.bing.com/new

4. 在现有的文献中，关于 LLM 的最小参数规模并没有形成正式共识，因为模型容量也与数据大小和总计算量有关。在本文献中，我们采用了一个略微宽松的 LLM 定义，并主要关注模型大小大于 10B 的语言模型。

例如 GPT-3 [55]、PaLM [56]、Galactica [34] 和 LLaMA [57]。具体而言，LLM 基于 Transformer 架构 [22] 构建，其中多头注意力层堆叠在非常深的神经网络中。现有的 LLM 主要采用与小语言模型类似的模型架构（即 Transformer）和预训练目标（即语言建模）。作为主要区别，LLM 大幅扩展了模型大小、预训练数据和总计算量（若干数量级），可以更好地根据上下文（即提示）理解自然语言并生成高质量的文本。这一能力提升可以部分通过扩展定律来描述，即任务性能大致随着模型大小的增加而显著提高 [30]。然而，一些能力（例如上下文学习 [55]）是不可预测的，只有当模型大小超过一定水平时才能观察到（如下文所述）。

大语言模型的涌现能力：在文献中 [47]，LLM 的“涌现能力”被正式定义为“在小模型中不存在但在大模型中出现的能力”，这是区分 LLM 与以前的 PLM 最突出的特征之一。它进一步介绍了一个显著的特征，即当规模达到一定水平时，性能显著提高超过随机水平。类比地，这种涌现模式与物理学中的“相变”现象有着密切的联系 [47, 58]。原则上，涌现能力可以定义为与某些复杂任务相关的能力 [47, 59]，而我们更关注能够应用于解决各种任务的通用能力。这里，我们简要介绍三个代表性的 LLM 涌现能力（详见第 7.2节）。

• 上下文学习：上下文学习能力由 GPT-3 正式引入 [55]：假设提供给语言模型自然语言指令和/或多个任务演示，它可以通过完成输入文本的单词序列来为测试实例生成期望的输出，而无需额外的训练或梯度更新[5]。

• 指令遵循：通过使用自然语言描述的多任务数据集进行微调（称为指令微调），LLM 可以在同样使用指令形式化描述的未见任务上表现良好 [28, 61, 62]。通过指令微调，LLM 能够在没有使用显式示例的情况下遵循任务指令，从而具有更好的泛化能力。

• 逐步推理：对于小语言模型来说，通常难以解决涉及多个推理步骤的复杂任务，例如数学问题。然而，通过采用“思维链”推理策略 [32]，LLM 可以利用包含中间推理步骤的提示机制来解决这些任务，得出最终答案。这种能力被认为可能通过在代码上进行训练来获得 [32, 46]。

LLM 关键技术：经过漫长的发展，LLM 进化到了当前的状态——通用且有能力的学习者。在这个过程中，人们提出了许多重要的技术，大大提升了 LLM 的能力。在此，我们简要列举了几种重要的技术，这些技术（可能）是导致 LLM 成功的关键。

• 扩展：扩展是提升 LLM 模型能力的关键因素。作为最初的尝试，GPT-3 首先将模型大小增加到了一个极大的规模，达到了 1750 亿参数。之后，PaLM 进一步将参数规模提升到了新纪录的 5400 亿。正如之前所讨论的，大的模型尺寸是涌现能力的关键。然而，扩展不仅与模型大小有关，还与数据大小和总运算量有关 [33, 63]。最近的一项研究 [33] 讨论了在给定固定预算的情况下，模型大小、数据大小和总计算量三个方面之间的最佳调配。此外，预训练数据的质量在实现良好性能方面起着关键作用，因此在扩展预训练语料库时，数据收集和清洗策略非常重要。

• 训练：由于巨大的模型规模，成功训练一种能力强的 LLM 是非常具有挑战性的。分布式训练算法是学习 LLM 网络参数所必需的，其中通常联合使用各种并行策略。为了支持分布式训练，已经发布了几个优化框架来促进并行算法的实现和部署，例如 DeepSpeed [64] 和 Megatron-LM [65–67]。此外，优化技巧对于训练稳定性和模型性能也很重要，例如重新开始以克服训练损失激增 [56] 和混合精度训练 [68]。最近，GPT-4 [45] 提出开发特殊的基础设施和优化方法，可靠地预测远小于大模型的小模型的性能。

• 能力引导：在大规模语料库上预训练之后，LLM 具备了作为通用任务求解器的潜在能力。然而，当 LLM 执行一些特定任务时，这些能力可能不会显式地展示出来。作为技术手段，设计合适的任务指令或具体的上下文学习策略可以激发这些能力。例如，通过包含中间推理步骤的思维链提示已被证明对解决复杂的推理任务有用。此外，我们还可以使用自然语言表达的任务描述对 LLM 进行指令微调，以提高 LLM 在未见过任务上的泛化能力。然而，这些技术主要对应于 LLM 的涌现能力，可能对小语言模型的效果不同。

• 对齐微调：由于 LLM 被训练来捕捉预训练语料库的数据特征（包括高质量和低质量的数据），它们可能会为人类生成有毒、偏见甚至有害的内容。因此，有必要使 LLM 与人类价值观保持一致，例如有用性、诚实性和无害性。为此，InstructGPT [61] 设计了一种有效的微调方法，使 LLM 能够按照期望的指令进行操作，其中利用了使用人类反馈的强化学习技术 [61, 69]。它将人类纳入训练循环中，采用精心设计的标注策略。ChatGPT 实际上采用类似于 InstructGPT 的技术，在产生高质量、无害的回答（例如拒绝回答侮辱性问题）方面表现出很强的对齐能力。

• 工具操作：从本质上讲，LLM 是基于海量纯文本语料库进行文本生成训练的，因此在那些不适合以文本形式表达的任务上表现不佳（例如数字计算）。此外，它们的能力也受限于预训练数据，例如无法获取最新信息。为了解决这些问题，最近提出了一种技术，即利用外部工具来弥补 LLM 的不足 [70, 71]。例如，LLM 可以利用计算器进行准确计算 [70]，利用搜索引擎检索未知信息 [71]。最近，ChatGPT 已经实现了使用外部插件（现有或新创建的应用程序）的机制[6]，这类似于 LLM 的“眼睛和耳朵”。这种机制可以广泛扩展 LLM 的能力范围。

此外，许多其他因素（例如硬件升级）也对 LLM 的成功做出了贡献。但是，我们主要讨论在开发 LLM 方面的主要技

5. 一些最近的研究 [60] 表明，上下文学习通过注意力机制隐式地执行元优化。

6. https://openai.com/blog/chatgpt-plugins

术方法和关键发现。

3 大语言模型资源

考虑到技术问题的挑战和计算资源的巨大需求，开发或再现大语言模型绝非易事。一种可行的方法是在现有的大语言模型的基础上进行开发，即重复使用公开可用的资源进行增量开发或实验研究。在本节中，我们简要整理了用于开发大语言模型的公开可用的资源，包括公开的模型检查点（或 API）、语料库和算法库。

3.1 公开可用的模型检查点或 API

考虑到模型预训练的巨大成本，训练良好的模型检查点对于研究组织开展大语言模型的研究和开发至关重要。由于参数规模是使用大语言模型时需要考虑的关键因素，为了帮助用户根据其资源预算确定适当的研究内容，我们将这些公开模型分为两个规模级别（百亿参数量级别和千亿参数量级别）。此外，也可以直接使用公开的 API 执行推理任务，而无需在本地运行模型。接下来，我们对公开可用的模型检查点和 API 进行介绍。

百亿参数量级别的模型：这类模型的参数规模除了 LLaMA（最大版本 65B 参数）和 NLLB（最大版本 54.5B 参数），大多在 10B 至 20B 之间。这一参数范围内的模型包括 mT5 [73]、PanGu-*alpha*[74]、T0 [28]、GPT-NeoX-20BGPT-NeoX-20B [77]、CodeGen [76]、UL2 [79]、Flan-T5 [83] 和 mT0 [84] 等。其中，Flan-T5（11B 版本）可以作为研究指令微调的首选模型，因为它从三个方面探索了指令微调 [83]：增加任务数量、扩大模型规模和使用链式思维提示数据进行微调。CodeGen（11B 版本）是一个为生成代码设计的自回归语言模型，可用作探索代码生成能力的候选模型，其提出了一个新的基准测试 MTPB [76]，专门用于多轮程序合成，由 115 个专家生成的问题组成，为了解决这些问题，需要大语言模型获得足够的编程知识（例如数学、数组操作和算法）。对于多语言任务，mT0（13B 参数版本）可能是一个比较好的候选模型，因为它在多语言任务中使用多语言提示进行了微调。此外，对于中文的下游任务，PanGu-α [74] 具有较好的表现，特别是在零样本或小样本的设置下，该模型基于深度学习框架 MindSpore [104] 开发，拥有多个参数版本（最大版本 200B 参数），而最大的公开版本只有 13B 参数。此外，作为最近发布的模型，LLaMA（65B 版本） [57] 在与指令遵循的任务中展现了卓越的性能。由于其开放性和有效性，LLaMA 引起了研究界的广泛关注，许多工作 [105–108] 致力于微调或继续训练其不同的模型版本以实现新模型或工具的开发。百亿参数量级别的模型通常需要数百甚至上千个 GPU 或 TPU。例如，GPT-NeoX-20B 使用了 12 个微服务器，每个服务器配备了 8 个 NVIDIA A100-SXM4-40GB GPU，LLaMA 使用了 2048 个 A100-80G GPU。为了准确估计所需的计算资源，我们还是建议使用衡量涉及计算量的指标，例如计算 FLOPS（每秒浮点数运算次数） [30]。

千亿参数量级别的模型：在这类模型中，只有少数几个模型进行了公开发布。其中，OPT [80]、OPT-IML [85]、BLOOM [68] 和 BLOOMZ [84] 的参数量几乎与 GPT-3（175B 版本）大致相同，而 GLM [82] 和 Galactica [34] 的参数数量分别为 130B 和 120B。其中，OPT（175B 版本）专注于复现和开源，旨在使研究人员能够进行大规模可重复研究。对于跨语言泛化研究，可以将 BLOOM（176B 版本）和 BLOOMZ（176B 版本）用作基础模型，因为其在多语言语言建模任务中具有较好的能力。在这些模型中，OPT-IML 进行了指令微调，是研究指令调整效果的较好选择。千亿参数量级别的模型通常需要数千个 GPU 或 TPU 进行训练。例如，OPT（175B 版本）使用了 992 个 A100-80GB GPU，GLM（130B 版本）使用了 96 个 NVIDIA DGX-A100（8x40G）GPU 节点集群。

大语言模型的公共 API：相较于直接使用模型副本，API 提供了一种更方便的方式供普通用户使用大语言模型，使得用户无需在本地运行模型。作为使用大语言模型的代表性接口，GPT 系列模型的 API [45, 55, 61, 88] 已经广泛应用于学术界和工业界[7]。OpenAI 提供了七个主要的 GPT-3 系列模型接口：`ada`、`babbage`、`curie`、`davinci`（GPT-3 系列中最强大的版本）、`text-ada-001`、`text-babbage-001` 和 `text-curie-001`。其中前四个接口可以在 OpenAI 的主机服务器上进一步进行微调。`babbage`、`curie` 和 `davinci` 分别对应于 GPT-3 (1B)、GPT-3 (6.7B) 和 GPT-3 (175B) 模型 [55]。此外，还有两个与 Codex 有关的 API，分别称为 `code-cushman-001`（Codex (12B) 的强大多语言版本 [88]）和 `code-davinci-002`。GPT-3.5 系列包括一个基础模型 `code-davinci-002` 和三个增强版本，即 `text-davinci-002`、`text-davinci-003` 和 `gpt-3.5-turbo-0301`。值得注意的是，`gpt-3.5-turbo-0301` 是调用 ChatGPT 的接口。最近，OpenAI 还发布了与 GPT-4 相应的 API，包括 `gpt-4`、`gpt-4-0314`、`gpt-4-32k` 和 `gpt-4-32k-0314`。总体而言，API 接口的选择取决于具体的应用场景和响应需求。详细的用法可以在它们的项目网站上找到[8]。

3.2 常用语料库

与早期的小型预训练语言模型不同，大语言模型有着规模极大的参数量，需要更大量且内容广泛的训练数据。为满足这种需求，越来越多的可用于研究的训练数据集被发布到公共社区中。在本节，我们将简要总结了一些常用于训练大语言模型的语料库。基于它们的内容类型，我们将这些语料库分为六个组别进行介绍：Books、CommonCrawl、Reddit Links、Wikipedia、Code、Others。

7. https://platform.openai.com/docs/api-reference/introduction
8. https://platform.openai.com/docs/models/overview

表 1

近年来大型语言模型（指规模大于 10B 的模型）的统计数据，包括 Evaluation、Pre-train Data Scale（以 token 数量或存储大小表示）和 Hardware。在本表中，我们仅列举有公开论文介绍技术细节的大语言模型。这里，“Release Time”表示相应论文正式发布的日期。“Publicly Available”表示模型检查点可以公开获取，而“Closed Source”则相反。“Adaptation”指模型是否经过了后续微调：IT 表示指令微调，RLHF 表示人类反馈的强化学习。“Evaluation”表示模型是否在原始论文中评估了相应的能力：ICL 表示上下文学习，CoT 表示思维链。“*”表示最大的公开可用版本。

	Model	Release Time	Size (B)	Base Model	Adaptation		Pre-train Data Scale	Latest Data Timestamp	Hardware (GPUs / TPUs)	Training Time	Evaluation	
					IT	RLHF					ICL	CoT
Publicly Available	T5 [72]	Oct-2019	11	-	-	-	1T tokens	Apr-2019	1024 TPU v3	-	✓	-
	mT5 [73]	Oct-2020	13	-	-	-	1T tokens	-	-	-	✓	-
	PanGu-α [74]	Apr-2021	13*	-	-	-	1.1TB	-	2048 Ascend 910	-	✓	-
	CPM-2 [75]	Jun-2021	198	-	-	-	2.6TB	-	-	-	-	-
	T0 [28]	Oct-2021	11	T5	✓	-	-	-	512 TPU v3	27 h	✓	-
	CodeGen [76]	Mar-2022	16	-	-	-	577B tokens	-	-	-	✓	-
	GPT-NeoX-20B [77]	Apr-2022	20	-	-	-	825GB	-	96 40G A100	-	✓	-
	Tk-Instruct [78]	Apr-2022	11	T5	✓	-	-	-	256 TPU v3	4 h	✓	-
	UL2 [79]	May-2022	20	-	-	-	1T tokens	Apr-2019	512 TPU v4	-	✓	✓
	OPT [80]	May-2022	175	-	-	-	180B tokens	-	992 80G A100	-	✓	-
	NLLB [81]	Jul-2022	54.5	-	-	-	-	-	-	-	✓	-
	GLM [82]	Oct-2022	130	-	-	-	400B tokens	-	768 40G A100	60 d	✓	-
	Flan-T5 [83]	Oct-2022	11	T5	✓	-	-	-	-	-	✓	✓
	BLOOM [68]	Nov-2022	176	-	-	-	366B tokens	-	384 80G A100	105 d	✓	-
	mT0 [84]	Nov-2022	13	mT5	✓	-	-	-	-	-	✓	-
	Galactica [34]	Nov-2022	120	-	-	-	106B tokens	-	-	-	✓	✓
	BLOOMZ [84]	Nov-2022	176	BLOOM	✓	-	-	-	-	-	✓	-
	OPT-IML [85]	Dec-2022	175	OPT	✓	-	-	-	128 40G A100	-	✓	✓
	LLaMA [57]	Feb-2023	65	-	-	-	1.4T tokens	-	2048 80G A100	21 d	✓	-
	Pythia [86]	Apr-2023	12	-	-	-	300B tokens	-	256 40G A100	-	✓	-
Closed Source	GPT-3 [55]	May-2020	175	-	-	-	300B tokens	-	-	-	✓	-
	GShard [87]	Jun-2020	600	-	-	-	1T tokens	-	2048 TPU v3	4 d	-	-
	Codex [88]	Jul-2021	12	GPT-3	-	-	100B tokens	May-2020	-	-	✓	-
	ERNIE 3.0 [89]	Jul-2021	10	-	-	-	375B tokens	-	384 V100	-	✓	-
	Jurassic-1 [90]	Aug-2021	178	-	-	-	300B tokens	-	800 GPU	-	✓	-
	HyperCLOVA [91]	Sep-2021	82	-	-	-	300B tokens	-	1024 A100	13.4 d	✓	-
	FLAN [62]	Sep-2021	137	LaMDA	✓	-	-	-	128 TPU v3	60 h	✓	-
	Yuan 1.0 [92]	Oct-2021	245	-	-	-	180B tokens	-	2128 GPU	-	✓	-
	Anthropic [93]	Dec-2021	52	-	-	-	400B tokens	-	-	-	✓	-
	WebGPT [71]	Dec-2021	175	GPT-3	-	✓	-	-	-	-	✓	-
	Gopher [59]	Dec-2021	280	-	-	-	300B tokens	-	4096 TPU v3	920 h	✓	-
	ERNIE 3.0 Titan [94]	Dec-2021	260	-	-	-	300B tokens	-	2048 V100	28 d	✓	-
	GLaM [95]	Dec-2021	1200	-	-	-	280B tokens	-	1024 TPU v4	574 h	✓	-
	LaMDA [96]	Jan-2022	137	-	-	-	2.81T tokens	-	1024 TPU v3	57.7 d	-	-
	MT-NLG [97]	Jan-2022	530	-	-	-	270B tokens	-	4480 80G A100	-	✓	-
	AlphaCode [98]	Feb-2022	41	-	-	-	967B tokens	Jul-2021	-	-	-	-
	InstructGPT [61]	Mar-2022	175	GPT-3	✓	✓	-	-	-	-	✓	-
	Chinchilla [33]	Mar-2022	70	-	-	-	1.4T tokens	-	-	-	✓	-
	PaLM [56]	Apr-2022	540	-	-	-	780B tokens	-	6144 TPU v4	-	✓	✓
	AlexaTM [99]	Aug-2022	20	-	-	-	1.3T tokens	-	128 A100	120 d	✓	✓
	Sparrow [100]	Sep-2022	70	-	-	✓	-	-	64 TPU v3	-	✓	-
	WeLM [101]	Sep-2022	10	-	-	-	300B tokens	-	128 A100 40G	24 d	✓	-
	U-PaLM [102]	Oct-2022	540	PaLM	-	-	-	-	512 TPU v4	5 d	✓	✓
	Flan-PaLM [83]	Oct-2022	540	PaLM	✓	-	-	-	512 TPU v4	37 h	✓	✓
	Flan-U-PaLM [83]	Oct-2022	540	U-PaLM	✓	-	-	-	-	-	✓	✓
	GPT-4 [45]	Mar-2023	-	-	✓	✓	-	-	-	-	✓	✓
	PanGu-Σ [103]	Mar-2023	1085	PanGu-α	-	-	329B tokens	-	512 Ascend 910	100 d	✓	-

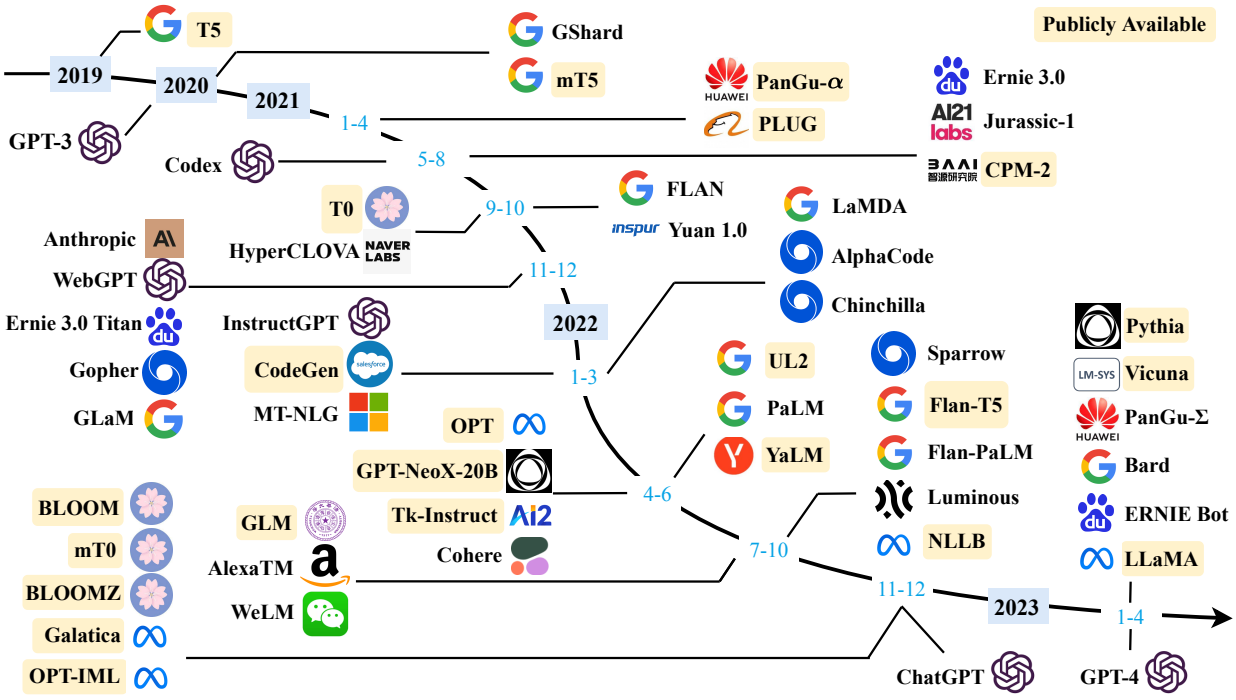


图 1. 最近几年现有的大型语言模型（大小超过 10B）的时间轴。时间轴主要是根据模型技术论文的发布日期（例如提交到 arXiv 的日期）建立的。如果没有相应的论文，我们将模型的日期设置为其公共发布或公告的最早时间。我们用黄色标记具有公开可用模型检查点的 LLM。由于图表空间的限制，我们仅包括公开报告了评估结果的 LLM。

表 2
常用语料库信息表。

Corpora	Size	Source	Latest Update Time
BookCorpus [109]	5GB	Books	Dec-2015
Gutenberg [110]	-	Books	Dec-2021
C4 [72]	800GB	CommonCrawl	Apr-2019
CC-Stories-R [111]	31GB	CommonCrawl	Sep-2019
CC-NEWS [27]	78GB	CommonCrawl	Feb-2019
REALNEWs [112]	120GB	CommonCrawl	Apr-2019
OpenWebText [113]	38GB	Reddit links	Mar-2023
Pushift.io [114]	-	Reddit links	Mar-2023
Wikipedia [115]	-	Wikipedia	Mar-2023
BigQuery [116]	-	Codes	Mar-2023
the Pile [117]	800GB	Other	Dec-2020
ROOTS [118]	1.6TB	Other	Jun-2022

Books： BookCorpus [109] 是之前小规模模型（如 GPT [119] 和 GPT-2 [26]）中常用的数据集，包括超过 11,000 本电子书，涵盖广泛的主题和类型（如小说和传记）。另一个大规模的书籍语料库是 Gutenberg [110]，它有超过 70,000 本文学作品，包括小说、散文、诗歌、戏剧、历史、科学、哲学和其他公共领域的作品。它是目前最大的开源书籍集合之一，被用于训练 MT-NLG [97] 和 LLaMA [57]。至于在 GPT-3 [55] 中使用的 Books1 [55] 和 Books2 [55]，它们比 BookCorpus 要大得多，但目前尚未公开发布。

CommonCrawl： CommonCrawl ([120]) 是最大的开源网络爬虫数据库之一，容量达到了百万亿字节级别，已经被广泛运用于训练大预言模型。由于整个数据集非常大，因此现有的研究主要在特定时间段内从中提取网页子集。然而，由于网络数据中存在大量的噪音和低质量信息，因此使用前需要进行数据预处理。目前有四个较为常用的基于 CommonCrawl 的过滤数据集：C4 ([72])，CC-Stories ([111])，CC-News ([27])，和 RealNews ([112])。C4 包括五个变种[9]，即 en (806G)，en.noclean (6T)，realnewslike (36G)，webtextlike (17G) 和 multilingual (38T)。其中，*en* 版本被用于预训练 T5 ([72])，LaMDA ([96])，Gopher ([59])，和 UL2 ([79])。而 multilingual C4，又被称为 mC4，被用于 mT5 ([73]) 的预训练。CC-Stories (31G) 是由 CommonCrawl 数据的子集组成，内容以故事的形式展示。然而，CC-Stories 的原始数据现在已不可用，因此在表2中是一个复现版本 *CC-Stories-R* ([121])。此外，还有两个从 CommonCrawl 中提取的两个新闻语料库：REALNEWS (120G) 和 CC-News (76G)，也常用作预训练数据。

Reddit Links： Reddit 是一个社交媒体平台，用户可以在上面提交链接和帖子，其他人可以通过“赞同”或“反对”投票。高赞的帖子通常被认为对多数用户是有帮助的，可以用来创建高质量的数据集。WebText [26] 就是一个著名的基于 Reddit 的

9. https://www.tensorflow.org/datasets/catalog/c4

语料库，它由 Reddit 上高赞的链接组成，但尚未公开。作为替代，有一个易于获取的开源替代品叫做 OpenWebText [113]。另一个从 Reddit 中提取的语料库是 PushShift.io [114]，它是一个实时更新的数据集，包括自 Reddit 创建以来的历史数据。Pushshift 不仅提供每月的数据转储，还提供有用的实用工具，支持用户搜索、总结和对整个数据集进行初步统计分析。这使得用户可以轻松地收集和处理 Reddit 数据。

Wikipedia： Wikipedia [115] 是一个在线百科全书，包含大量高质量的文章，涵盖各种主题。其中大部分文章都采用解释性写作风格（并支持引用），覆盖了多种不同语言和广泛的知识领域。通常来说，Wikipedia 英语版本被广泛应用于大多数大语言模型（*e.g.,* GPT-3 [55]，LaMDA [96] 和 LLaMA [57]）。它还提供多种语言版本，因此可以在多语言环境下使用。

Code： 为了收集代码数据，现有工作主要是从互联网上爬取有开源许可证的代码。代码数据有两个主要来源：包括开源许可证的公共代码库（*e.g.,* GitHub）和与代码相关的问答平台（*e.g.,* StackOverflow）。Google 公开发布了 BigQuery 数据集 [116]，其中包括各种编程语言的大量开源许可证代码片段，是一个典型的代码数据集。CodeGen 使用的 BIGQUERY [76] 是 BigQuery 数据集的一个子集，用于训练多语言版本的 CodeGen（CodeGen-Multi）。

Others： The Pile [117] 是一个大规模、多样化、开源的文本数据集，有超过 800GB 数据，内容包括书籍、网站、代码、科学论文和社交媒体平台等。它由 22 个多样化的高质量子集构成。The Pile 数据集被广泛应用于不同参数规模的模型，如 GPT-J（6B）[122]、CodeGen（16B）[76] 和 Megatron-Turing NLG（530B）[97]。此外，ROOTS[118] 由各种较小的数据集（总共 1.61 TB 的文本）组成，覆盖 59 种不同的语言（包括自然语言和编程语言），它被用于训练 BLOOM [68]。

实际上，为了预训练大语言模型，通常需要混合使用不同的数据源（见图 2），而不是单一的语料库。因此，现有的研究通常混合几个现成的数据集（如 C4、OpenWebText 和 the Pile 等），然后进行进一步的处理以获取预训练语料库。此外，为了训练适用于特定应用的大语言模型，从相关源（如 Wikipedia 和 BigQuery）提取数据以丰富预训练数据中的相应信息也很重要。

为了快速了解现有大语言模型使用的数据来源，我们介绍三个代表性大语言模型的预训练语料库：

• *GPT-3*（175B）[55] 是在混合数据集（共 300B 词）上进行训练的，包括 CommonCrawl[120]、WebText2 [55]、Books1 [55]、Books2 [55] 和 Wikipedia [115]。

• *PaLM*（540B）[56] 使用了一个由社交媒体对话、过滤后的网页、书籍、Github、多语言维基百科和新闻组成的预训练数据集，共包含 780B 词。

• *LLaMA*[57] 从多个数据源中提取训练数据，包括 CommonCrawl、C4[72]、Github、Wikipedia、书籍、ArXiv 和 StackExchange。LLaMA（6B）和 LLaMA（13B）的训练数据大小为 1.0T 词，而 LLaMA（32B）和 LLaMA（65B）使用了 1.4T 词。

3.3 算法库资源

在这部分，我们简要介绍了一些可用于开发大语言模型的算法库。

• **Transformers** [123] 是一个使用 Transformer 架构构建模型的开源 Python 库，由 Hugging Face 开发和维护。它具有简单和用户友好的 API，使得使用和定制各种预训练模型变得容易。它是一个功能强大的库，拥有庞大而活跃的用户和开发者社区，他们定期更新和改进模型和算法。

• **DeepSpeed**[64] 是由 Microsoft 开发的深度学习优化库（与 PyTorch 兼容），已用于训练多个大语言模型，例如 MT-NLG[97] 和 BLOOM [68]。它提供了各种分布式训练优化技术的支持，例如内存优化（ZeRO 技术、梯度检查点）和管道并行。

• **Megatron-LM** [65–67] 是由 NVIDIA 开发的深度学习库，用于训练大语言模型。它提供了丰富的分布式训练优化技术，包括模型和数据并行、混合精度训练和 FlashAttention。这些优化技术可以大大提高训练效率和速度，并实现 GPU 间的高效分布式训练。

• **JAX** [124] 是由 Google 开发的用于高性能机器学习算法的 Python 库，允许用户在带有硬件加速（例如 GPU 或 TPU）的情况下进行数组的高效运算。它可以在各种设备上进行高效计算，还支持自动微分和即时编译等特色功能。

• **Colossal-AI**[125] 是由 HPC-AI Tech 开发的用于训练大规模人工智能模型的深度学习库。它基于 PyTorch 实现，并支持丰富的并行训练策略。此外，它还可以使用 PatrickStar[126] 提出的方法优化异构内存管理。最近，使用 Colossal-AI 基于 LLaMA [57] 开发的类 ChatGPT 模型 ColossalChat [108]（7B 和 13B 版本）已经公开发布。

• **BMTrain**[127] 是由 OpenBMB 开发的用于以分布式方式训练大规模参数模型的高效库，强调代码简洁、低资源占用和高可用性。BMTrain 已经将一些常见的大语言模型（如 Flan-T5[83] 和 GLM [82]）迁移到到其 ModelCenter 中，用户可以直接使用这些模型。

• **FastMoE** [128] 是一种专门用于 MoE（即混合专家）模型的训练库。它基于 PyTorch 开发，注重效率和用户友好性。FastMoE 简化了将 Transformer 模型转换为 MoE 模型的过程，并支持数据并行和模型并行训练。

除了上述的库资源外，其他深度学习框架（例如 PyTorch [129]，TensorFlow [130]，MXNet [131]，PaddlePaddle [132]，MindSpore [104] 和 OneFlow [133]）也提供了并行算法支持，这些算法通常用于训练大规模模型。

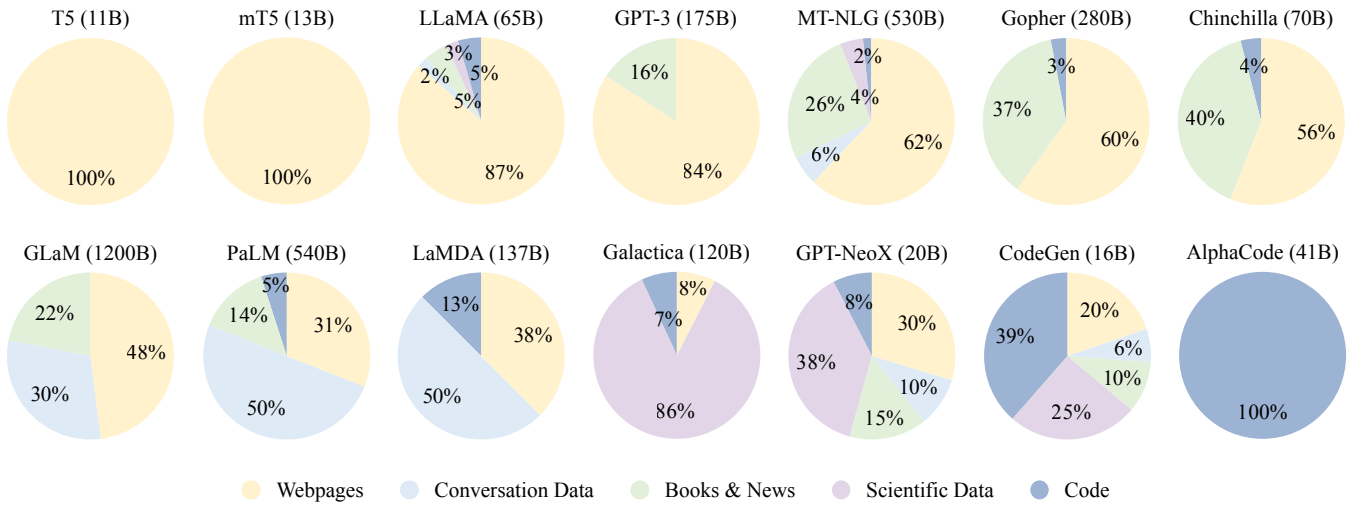


图 2. 现有大语言模型预训练数据中各种数据来源的比率。

4 预训练

预训练是大语言模获取能力的基础。通过在大规模语料库上进行预训练，大语言模型可以获得基本的语言理解和生成能力 [55, 56]。在这个过程中，预训练语料库的规模和质量对于大语言模型获得强大的能力至关重要。此外，为了有效地预训练大语言模型，也需要设计好模型架构、加速方法和优化技术。接下来，我们首先在第 4.1节讨论数据收集和处理，然后在第 4.2节介绍常用的模型架构，最后在第 4.3节介绍稳定高效地优化大语言模型的训练技巧。

4.1 数据收集

相比小规模语言模型，大语言模型更需要高质量数据来预训练模型，并且它们的模型容量很大程度上依赖于预训练语料库及其预处理方式。在这一部分，我们讨论预训练数据的收集和处理，包括数据来源、预处理方法以及预训练数据如何影响大语言模型的性能等重要分析。

4.1.1 数据来源

为了开发一个能力强大的大语言模型，收集大量各种来源的自然语言语料库至关重要。现有的大语言模型主要混合各种公共文本数据集作为预训练语料库。图 2展示了一些代表性大语言模型的预训练数据来源的分布情况。

预训练语料库的来源可以广义地分为两种类型：通用文本数据和专用文本数据。通用数据，如网页、书籍和对话文本等，由于其规模大、多样性强且易于获取的特点，被大多数大语言模型所利用 [55, 56, 80]，这可以增强大语言模型的语言建模和泛化能力。鉴于大语言模型所展现出的惊人泛化能力，也有研究将预训练语料库扩展到更专用的数据集，如多语言数据、科学数据和代码等，来赋予大语言模型解决专用任务的能力 [34, 56, 76]。接下来，我们将描述这两种类型的预训练数据来源以及它们对大语言模型的影响。关于常用语料库的详细介绍，可以参考第 3.2节。

通用文本数据： 如图 2 所示，绝大多数的大语言模型采用了通用的预训练数据，比如网页、书籍和对话文本等，这些数据源提供了丰富的文本资源，涉及了多种主题。接下来，我们简要总结三种重要的通用数据。

- 网页：随着互联网的普及，多种多样的数据被创造出来，这些丰富的数据使得大语言模型能够获得多样化的语言知识并增强大语言模型的泛化能力 [26, 72]。为了方便使用这些数据资源，之前的工作从网络中爬取了大量的数据，如 CommonCrawl [120]。然而，这些爬取的网络数据往往同时包含高质量的文本，如维基百科，和低质量的文本，如垃圾邮件，因此过滤和处理网页以提高数据质量非常重要。
- 对话文本：对话数据可以增强大语言模型的对话能力 [80]，可能也提高了大语言模型在问答任务上的表现 [56]。研究人员可以利用公共对话语料库的子集（如 PushShift.io Reddit 语料库）[114, 134]，或从在线社交媒体收集对话数据。由于在线对话数据通常涉及多个参与者之间的讨论，因此一种有效的处理方式是将对话转换成树形结构，其中每句话与回应它的话语相连。通过这种方式，多方之间的的对话树可以被划分为预训练语料库中的多个子对话。然而，过度引入对话数据来训练大语言模型可能会导致一个潜在的风险 [80]：声明性指令和直白的疑问会被错误地认为是对话的开始，从而导致指令的有效性下降。
- 书籍：与其他语料库相比，书籍提供了更正式的长文本，这对于大语言模型学习语言知识、建模长期依赖关系和生成连贯的文本可能带来了好处。为了获得开源书籍数据，现有的研究通常采用 Books3 和 Bookcorpus2 数据集，这些数据集可以在 Pile 数据集中获得 [117]。

专用文本数据： 专用数据集对于提高大语言模型在特定下游任务中的能力非常有用。接下来，我们介绍三种专用数据类型。

- 多语言文本：除了在单目标语言上进行训练外，整

合多语言语料库可以增强多语言的理解和生成能力。例如，BLOOM [68] 和 PaLM [56] 在其预训练语料库中收集了包含 46 种和 122 种语言的多语言数据。这些模型在多语言任务中展现出了出色的性能，例如翻译、多语言摘要和多语言问答，并且相比于在目标语言上微调的最先进的模型具有可比或更好的性能。

• 科学文本：科学出版物的不断增长见证了人类对科学的探索。为了增强大语言模型对科学知识的理解 [34, 135]，可以将科学语料库纳入模型的预训练语料 [34, 135]。通过在大量科学文本上进行预训练，大语言模型可以在科学和推理任务中取得出色的性能 [136]。为了构建科学语料库，现有的工作主要收集 arXiv 论文、科学教材、数学网页和其他相关的科学资源。由于科学领域数据的复杂性，例如数学符号和蛋白质序列，通常需要特定的标记化和预处理技术将这些不同格式的数据转换为可以被语言模型处理的统一形式。

• 代码：程序编写在学术界得到了广泛的研究 [88, 137–140]，特别是使用训练于代码的预训练语言模型 [122, 141]。然而，对于这些预训练语言模型（如 GPT-J [122]），生成高质量和准确的程序仍然具有挑战性。最近的研究 [88, 140] 发现，在大量的代码语料库上预训练打语言模型可以显著提高编写程序的质量。编写的程序可以成功通过专家设计的单元测试用例 [88] 或解决竞赛编程问题 [98]。一般来说，常用于预训练大语言模型的代码语料库有两种来源。第一种来源是来自编程问答社区（如 Stack Exchange） [142, 143]。第二种来源是来自公共软件仓库，例如 GitHub [76, 88, 140]，它们收集了代码数据（包括注释和文档字符串）以供利用。与自然语言文本相比，代码以编程语言的格式呈现，对应着长距离依赖和准确的执行逻辑 [144]。最近的一项研究 [46] 还推测，训练代码可能是复杂推理能力（如思维链能力 [32]）的来源。此外，将推理任务格式化为代码也可以帮助大语言模型生成更准确的结果 [144, 145]。

4.1.2 数据预处理

在收集大量文本数据后，对数据进行预处理，特别是消除噪声、冗余、无关和潜在有害的数据 [56, 59]，对于构建预训练语料库是必不可少的，因为这些数据可能会极大地影响大语言模型的能力和性能。在这部分中，我们将细致地回顾提高收集数据的质量的数据预处理策略 [59, 68, 95]。预处理大语言模型的预训练数据的典型流程已在图 3中说明。

质量过滤： 为删除收集到的语料库中的低质量数据，现有的工作通常采用两种方法：(1) 基于分类器的方法，和 (2) 基于启发式的方法。前一种方法基于高质量文本训练选择分类器，并利用它来识别和过滤低质量数据。通常，这些方法 [55, 56, 95] 使用高质量数据（如维基百科页面）作为正样本，采样候选数据作为负样本来训练二元分类器，并预测衡量每个数据示例质量的分数。然而，一些研究 [59, 95] 也发现，基于分类器的方法可能会删除方言、口语和社会语言的高质量文本，从而可能导致有偏的预训练语料库，并减少语料库的多样性。对于第二种方法，一些研究，如 BLOOM [68] 和 Gopher [59]，采用基于启发式的方法，通过设计一组精心设计的规则来消除低质量文本，这些规则可以总结如下：

- 基于语言的过滤：如果大语言模型主要用于某项语言的任务中，那么其他语言的文本可以被过滤掉。
- 基于度量的过滤：可以利用生成文本的评估度量，例如困惑度（perplexity），来检测和删除不自然的句子。
- 基于统计的过滤：可以利用语料库的统计特征，例如标点符号分布、符号与单词比率和句子长度，来衡量文本质量并过滤低质量数据。
- 基于关键词的过滤：基于特定的关键词集合，可以识别和删除文本中的噪声或无用元素，例如 HTML 标签、超链接、模板和攻击性词语。

去重： 现有的研究 [146] 发现，语料库中的重复数据会降低语言模型的多样性，可能导致训练过程不稳定，从而影响模型性能。因此，需要对预训练语料库进行去重处理。具体来说，可以在句子级、文档级和数据集级等不同粒度上去重。首先，在句子级别上，应删除包含重复单词和短语的低质量句子，因为它们可能会在语言建模中引入重复模式 [147]。在文档级别上，现有研究主要依靠文档之间的表层特征（例如单词和 n-gram 的重叠）重叠比率来检测和删除包含相似内容的重复文档 [57, 59, 68, 148]。此外，为了避免数据集污染问题，还必须通过从训练集中删除测试集可能出现的重复文本，来防止训练集和评估集之间的重叠 [56]。已经证明，这三个级别的去重都有助于改善大语言模型的训练 [56, 149]，在实践中应该共同使用这三个级别的去重。

隐私去除： 大多数预训练文本数据来自网络来源，包括涉及敏感或个人信息的用户生成内容，这可能增加隐私泄露的风险 [150]。因此，需要从预训练语料库中删除可识别个人信息 (*PII*)。一种直接有效的方法是采用基于规则的方法，例如关键字识别，来检测和删除 PII，例如姓名、地址和电话号码 [118]。此外，研究人员还发现，大语言模型在隐私攻击下的脆弱性可能归因于预训练语料库中存在的重复 PII 数据 [151]。因此，去重也可以在一定程度上降低隐私风险。

分词： 分词也是数据预处理的关键步骤。它的目的是将原始文本分割成词序列，随后用作大语言模型的输入。虽然直接利用已有的分词器是方便的（例如 OPT [80] 和 GPT-3 [55] 利用 GPT-2 [26] 的分词器），但是使用专门为预训练语料库设计的分词器可能会更加有效 [68]，特别是对于由多种领域、语言和格式组成的语料库。因此，最近的几个大语言模型使用 SentencePiece [152] 为预训练语料库训练定制化的分词器。同时利用字节级 *Byte Pair Encoding (BPE)* 算法 [153] 确保分词后信息不丢失 [56, 59]。然而需要注意的是，BPE 中的规范

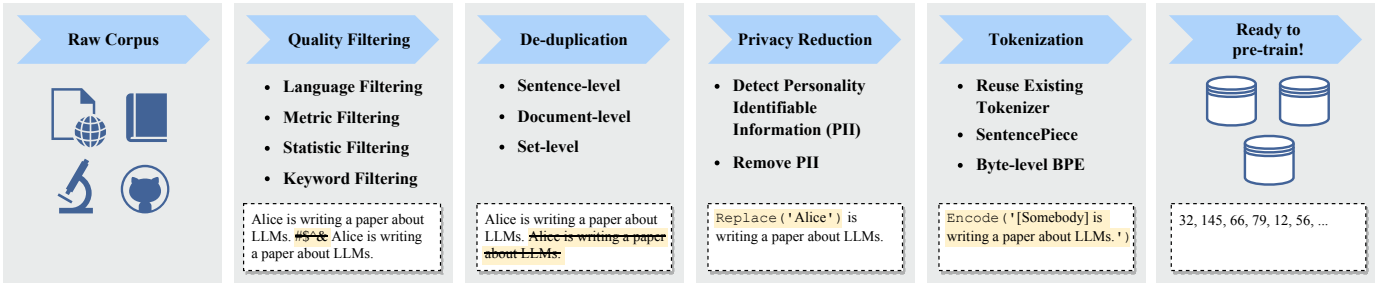


图 3. 一个典型的预处理预训练数据的流程图。

化技术，例如 NFKC [154]，可能会降低分词性能 [33, 59, 68]。

4.1.3 预训练数据对大语言模型的影响

与小规模的预训练语言模型不同，由于对计算资源的巨大需求，通常不可能对大语言模型进行多次预训练迭代。因此，在训练大语言模型之前构建一个准备充分的预训练语料库尤为重要。在这一部分中，我们将讨论预训练语料库的质量和分布会如何潜在地影响大语言模型的性能。

混合来源：正如前面所讨论的，来自不同领域或场景的预训练数据具有不同的语言特征或语义知识。通过在来自不同来源的文本数据上进行预训练，大语言模型可以获得广泛的知识，并可能展现出强大的泛化能力。当混合不同来源的数据时，需要仔细设置预训练数据的分布，因为这也可能会影响大语言模型在下游任务上的性能 [59]。Gopher [59] 对数据分布进行了消融实验，以检验混合来源对下游任务的影响。它在 LAMBADA 数据集 [155] 上的实验结果表明，增加书籍数据的比例可以提高模型从文本中捕捉长期依赖的能力，增加 C4 数据集 [72] 的比例则会提升它在 C4 验证数据集 [59] 上的性能。然而，单独训练过多的某个领域的数据会影响大语言模型在其他领域的泛化能力 [34, 59]。因此，建议研究人员应仔细确定预训练语料库中来自不同领域的数据的比例，以开发更符合其特定需求的大语言模型。读者可以参考图 2，了解和比较不同大语言模型的数据来源。

预训练数据的数量：为了预训练一个有效的大语言模型，收集足够的高质量数据以满足大语言模型的数据数量需求是很重要的。现有研究发现，随着大语言模型参数规模的增加，也需要更多的数据来训练模型 [33, 57]：与模型大小相关的数据大小也观察到了类似的扩展定律，与模型性能有关。Chinchilla [33] 表明，许多现有的大语言模型由于缺乏充足的预训练数据而受到次优训练的影响。通过进行广泛的实验，它进一步表明，在给定的计算预算下，采用相等规模的模型参数和训练标记是必要的。最近，LLaMA [57] 表明，随着更多的数据和更长时间的训练，较小的模型也可以实现良好的性能。因此，建议研究人员在扩展模型参数时，尤其要注意高质量数据的数量，以充分训练模型。

预训练数据的质量：现有的研究表明，对低质量的语料库进行预训练，如噪声、有毒和重复的数据，可能会损害模型的性能 [59, 146, 148, 151]。为了开发表现良好的大语言模型，收集的训练数据的数量和质量都是至关重要的。最近的研究，如 T5 [72]、GLaM [95] 和 Gopher [59]，已经研究了数据质量对下游任务性能的影响。通过比较在过滤和未过滤的语料库上训练的模型的性能，它们得到了相同的结论，即在清理后的数据上预训练大语言模型可以提高性能。更具体地说，数据的重复可能会导致“双下降现象”（指性能最初恶化，随后得到改善）[146, 156]，甚至会使训练过程不稳定 [146]。此外，已经表明，重复的数据会降低大语言模型从上下文中复制的能力，这可能进一步影响大语言模型在上下文学习中的泛化能力 [146]。因此，正如 [56, 59, 68] 所建议的，研究人员有必要仔细地对预训练语料库进行预处理（如在第 4.1.2节中所示），以提高训练过程的稳定性并避免影响模型性能。

4.2 架构

本节中，我们将回顾大语言模型的架构设计，包括主流架构、预训练目标和详细配置。表 3列出了几个具有公开细节的代表性大语言模型的模型。

4.2.1 主流架构

由于 Transformer 架构的出色并行性和容量，Transformer 架构已成为开发各种大语言模型的事实标准骨干，使得将语言模型扩展到数百亿或数千亿个参数成为可能 [22]。一般来说，现有大语言模型的主流架构可以大致分为三种类型，即编码器-解码器、因果解码器和前缀解码器。

编码器-解码器架构：传统 Transformer 模型是建立在编码器-解码器架构上的 [22]，由两个 Transformer 块分别作为编码器和解码器。编码器采用堆叠的多头自注意层对输入序列进行编码以生成其潜在表示，而解码器对这些表示进行交叉注意并自回归地生成目标序列。编码器-解码器 PLMs（例如 T5 [72] 和 BART [24]）在各种 NLP 任务上表现出有效性。目前，只有少数大语言模型是基于编码器-解码器架构构建的，例如 Flan-T5 [83]。有关架构选择的详细讨论将在第 4.2.4节中进行。

表 3
这里列出了几个具有公开配置细节的选定大语言模型的模型。其中，PE 表示位置编码，#L 表示层数，#H 表示注意力头数，d_{model} 表示隐藏状态的大小，而 MCL 表示训练期间的最大上下文长度。

Model	Category	Size	Normalization	PE	Activation	Bias	#L	#H	d_{model}	MCL
GPT3 [55]	Causal decoder	175B	Pre Layer Norm	Learned	GeLU	✓	96	96	12288	2048
PanGU- α [74]	Causal decoder	207B	Pre Layer Norm	Learned	GeLU	✓	64	128	16384	1024
OPT [80]	Causal decoder	175B	Pre Layer Norm	Learned	ReLU	✓	96	96	12288	2048
PaLM [56]	Causal decoder	540B	Pre Layer Norm	RoPE	SwiGLU	×	118	48	18432	2048
BLOOM [68]	Causal decoder	176B	Pre Layer Norm	ALiBi	GeLU	✓	70	112	14336	2048
MT-NLG [97]	Causal decoder	530B	-	-	-	-	105	128	20480	2048
Gopher [59]	Causal decoder	280B	Pre RMS Norm	Relative	-	-	80	128	16384	2048
Chinchilla [33]	Causal decoder	70B	Pre RMS Norm	Relative	-	-	80	64	8192	-
Galactica [34]	Causal decoder	120B	Pre Layer Norm	Learned	GeLU	×	96	80	10240	2048
LaMDA [96]	Causal decoder	137B	-	Relative	GeGLU	-	64	128	8192	-
Jurassic-1 [90]	Causal decoder	178B	Pre Layer Norm	Learned	GeLU	✓	76	96	13824	2048
LLaMA [57]	Causal decoder	65B	Pre RMS Norm	RoPE	SwiGLU	✓	80	64	8192	2048
GLM-130B [82]	Prefix decoder	130B	Post Deep Norm	RoPE	GeGLU	✓	70	96	12288	2048
T5 [72]	Encoder-decoder	11B	Pre RMS Norm	Relative	ReLU	×	24	128	1024	512

因果解码器架构：因果解码器架构采用单向注意力掩码，以确保每个输入标记只能关注过去的标记和它本身。输入和输出标记通过解码器以相同的方式进行处理。作为这种架构的代表性语言模型，GPT 系列模型 [26, 55, 119] 是基于因果解码器架构开发的。特别地，GPT-3 [55] 成功展示了这种架构的有效性，同时也展示了大语言模型惊人的上下文学习能力。有趣的是，GPT-1 [119] 和 GPT-2 [26] 没有展现出与 GPT-3 相同的卓越能力，表明了模型规模的扩大在增加这种模型架构的模型容量方面起到了重要作用。迄今为止，因果解码器已被广泛采用为各种现有大语言模型的体系结构，例如 OPT [80]、BLOOM [68] 和 Gopher [59]。注意，接下来讨论的因果解码器和前缀解码器都属于仅解码器体系结构。当提到“仅解码器结构”时，除非另有说明，否则主要是指现有文献中的因果解码器架构。

前缀解码器架构：前缀解码器架构（也称非因果解码器架构）修正了因果解码器的掩码机制，以使其能够对前缀标记执行双向注意力 [157]，并仅对生成的标记执行单向注意力。这样，与编码器-解码器架构类似，前缀解码器可以双向编码前缀序列并自回归地逐个预测输出标记，其中在编码和解码过程中共享相同的参数。实用的建议是不从头开始进行预训练，而是继续训练因果解码器，然后将其转换为前缀解码器以加速收敛 [29]，例如 U-PaLM [102] 是从 PaLM [56] 演化而来。基于前缀解码器的现有代表性大语言模型包括 GLM-130B [82] 和 U-PaLM [102]。

对于这三种类型的架构，我们也可以考虑通过专家混合（MoE）扩展它们，其中每个输入的一小部分神经网络权重被稀疏激活，例如 Switch Transformer [25] 和 GLaM [95]。已经证明，通过增加专家的数量或总参数大小，可以观察到显著的性能改进 [158]。

4.2.2 详细配置

自 Transformer [22] 推出以来，已经提出了各种改进方法来提高其训练稳定性、性能和计算效率。在这部分中，我们将讨论 Transformer 的四个主要部分，包括标准化、位置编码、激活函数、注意力和偏置的相应配置。

标准化：训练不稳定是预训练大语言模型的一个难题。为了缓解这个问题，层标准化 (Layer Norm, LN) [159] 被广泛应用于 Transformer 架构中。LN 的位置对大语言模型的性能至关重要。虽然最初的 Transformer [22] 使用后置 LN，但大多数大语言模型采用前置 LN 以实现更稳定的训练，尽管会带来一定的性能损失 [160]。基于前置 LN，Sandwich-LN [161] 在残差连接之前添加额外的 LN，以避免数值爆炸。然而，已有研究发现 Sandwich-LN 有时无法稳定大语言模型的训练，可能导致训练崩溃 [82]。最近，一些高级标准化技术被提出以作为 LN 的替代方案。由于 RMS Norm 在训练速度和性能方面的优越性 [162]，其在 Gopher [59] 和 Chinchilla [33] 中被采用。与 LN 相比，DeepNorm [163] 已经表现出更好的训练稳定性，和后标准化一起被 GLM-130B 采用。此外，在嵌入层后添加额外的 LN 也可以稳定大语言模型的训练。然而，这往往会导致显著的性能下降 [164]，在一些最近的大语言模型中已经被移除 [68]。

激活函数：为了获得良好的性能，在前馈网络中也需要设置合适的激活函数。在现有的大语言模型中，广泛使用 GeLU 激活函数 [165]。此外，在最新的大语言模型 (*e.g.,* PaLM 和 LaMDA) 中，也使用了 GLU 激活函数的变体 [166, 167]，特别是 SwiGLU 和 GeGLU 变体，在实践中通常可以获得更好的性能 [168]。然而，与 GeLU 相比，它们在前馈网络中需要额外的参数（约 50%）[164]。

位置编码：由于 Transformer 中的自注意模块是置换等变的，因此需要使用位置编码来注入绝对或相对位置信息以建模序列。在经典的 Transformer [22] 中有两种绝对位置编码的变体，即正弦函数和学习的位置编码，后者通常在大语言模型中使用。与绝对位置编码不同，相对位置编码根据键和查询之间的偏移量生成嵌入 [72]，因此它可以在训练中看到的长度范围之外的更长序列上表现良好，即外推 [169]。ALiBi [169] 使用基于键和查询之间距离的惩罚来偏置注意力分数。实证结果表明，它比其他位置编码具有更好的零样本泛化能力和更强的外推能力 [29]。此外，通过基于绝对位置设置特定的旋转矩阵，RoPE [170] 中的键和查询之间的分数可以使用相对位置信息计算，这对于建模长序列是有用的。因此，RoPE 已经被广泛应用于一些最新的大语言模型 [56, 57, 82]。

注意力机制和偏差：除了原始 Transformer 中的全自注意力机制 [22]，GPT-3 采用了更低计算复杂度的稀疏注意力机制，即分解注意力 [55, 171]。为了有效且高效地建模更长的序列，研究者们探索了引入特殊的注意力模式 [172, 173] 或考虑 GPU 内存访问（即 FlashAttention [174]）。此外，与原始 Transformer 一样，大多数大语言模型在每个线性层和层标准化中保留了偏置。然而，在 PaLM [56] 和 Galactica [34] 中，偏置被移除。研究表明，对于大语言模型来说，去除偏置可以增强训练的稳定性 [56]。

综合上述讨论，我们总结了现有文献中的详细配置建议。为了更强的泛化能力和训练稳定性，建议选择预先的 RMS 标准化进行层标准化，并选择 SwiGLU 或 GeGLU 作为激活函数。此外，在位置编码方面，RoPE 或 ALiBi 是更好的选择，因为它们在长序列上表现更好。

4.2.3 预训练任务

预训练在将大规模语料库中的通用知识编码到巨大的模型参数中起着关键作用。对于训练 LLMs，有两个常用的预训练任务，即语言建模和去噪自编码。

语言模型：语言模型任务（LM）是预训练仅包含解码器的大语言模型（如 GPT3 [55] 和 PaLM [56]）最常用的目标。给定一个标记序列 $\mathbf{x} = \{x_1, \ldots, x_n\}$，LM 任务旨在基于序列中前面的标记 $x_{<i}$，自回归地预测目标标记 x_i。通常的训练目标是最大化以下似然函数：

$$\mathcal{L}_{LM}(\mathbf{x}) = \sum_{i=1}^{n} \log P(x_i|x_{<i}). \tag{1}$$

由于大多数语言任务可以基于输入的预测问题来解决，这些仅包含解码器的大语言模型可能具有优势，可以隐式地学习如何以统一的 LM 方式完成这些任务。一些研究还表明，仅包含解码器的大语言模型可以通过自回归地预测下一个标记而自然地转移到某些任务中，而无需微调 [26, 55]。LM 的一个重要变体是前缀语言模型任务，它是为预训练具有前缀解码器架构的模型设计的。在计算前缀语言模型的损失时，不使用随机选择的前缀内的标记。由于模型预训练涉及的序列中涉及的标记较少，因此在使用相同数量的预训练标记时，前缀语言模型的性能往往略低于传统语言模型任务 [29]。

去噪自编码：除了传统的 LM 之外，去噪自编码任务（DAE）也被广泛用于预训练语言模型 [24, 72]。DAE 任务的输入 $\mathbf{x}_{\setminus\tilde{\mathbf{x}}}$ 是一些有随机替换区间的损坏文本。然后，语言模型被训练以恢复被替换的标记 $\tilde{\mathbf{x}}$。形式上，DAE 的训练目标如下：

$$\mathcal{L}_{DAE}(\mathbf{x}) = \log P(\tilde{\mathbf{x}}|\mathbf{x}_{\setminus\tilde{\mathbf{x}}}). \tag{2}$$

然而，DAE 任务在实现上似乎比 LM 任务更为复杂。因此，它并没有被广泛用于预训练大型语言模型。采用 DAE 作为预训练目标的现有大语言模型包括 T5 [72] 和 GLM-130B [82]。这些模型主要通过自回归地恢复替换区间来进行训练。

4.2.4 总结与讨论

选择架构和预训练任务可能会对大语言模型的归纳偏差产生不同影响，从而导致不同的模型容量。在本部分中，我们总结了现有文献中关于这个问题的一些重要发现或讨论。

- 通过使用语言模型（LM）目标进行预训练，因果解码器架构似乎可以实现更优越的零样本和少样本泛化能力。现有研究表明，在没有进行多任务微调的情况下，因果解码器比其他架构具有更好的零样本性能 [29]。GPT-3 [55] 的成功证明了大因果解码器模型可以成为一个很好的少样本学习器。此外，第 5节中讨论的指令调整和对齐调整已被证明可以进一步增强大因果解码器模型的能力 [61, 62, 83]。
- 因果解码器中已经广泛观察到了扩展定律。通过扩展模型大小、数据集大小和总计算量，可以大幅提高因果解码器的性能 [30, 55]。因此，通过扩展已成为提高因果解码器模型容量的重要策略。然而，对于编码器-解码器模型的更详细研究仍然缺乏，需要更多的努力来研究大规模编码器-解码器模型的性能。此外，对于具有复杂交叉注意力掩码策略的编码器-解码器模型进行扩展在实践中更加困难。

当前仍需要更多架构和预训练任务的研究，以分析架构和预训练任务的选择如何影响大语言模型的容量，特别是对于编码器-解码器架构。除了主要架构外，大语言模型的详细配置也值得关注，这已在第 4.2.2节中讨论过。

4.3 模型训练

在这一部分中，我们回顾了训练大语言模型（LLMs）的重要设置、技巧或诀窍。

4.3.1 优化设置

为了进行大语言模型的参数优化，我们介绍了批量训练、学习率、优化器和训练稳定性的常用设置。

批量训练：对于语言模型的预训练，现有的研究通常将批量大小设置为较大的数字（例如 8,196 个样例或 1.6M 个标记），

表 4
各种现有大语言模型的详细优化设置。

Model	Batch Size (#tokens)	Learning Rate	Warmup	Decay Method	Optimizer	Precision Type	Weight Decay	Grad Clip	Dropout
GPT3 (175B)	32K→3.2M	6×10^{-5}	yes	cosine decay to 10%	Adam	FP16	0.1	1.0	-
PanGu-α (200B)	-	2×10^{-5}	-	-	Adam	-	0.1	-	-
OPT (175B)	2M	1.2×10^{-4}	yes	manual decay	AdamW	FP16	0.1	-	0.1
PaLM (540B)	1M→4M	1×10^{-2}	no	inverse square root	Adafactor	BF16	lr^2	1.0	0.1
BLOOM (176B)	4M	6×10^{-5}	yes	cosine decay to 10%	Adam	BF16	0.1	1.0	0.0
MT-NLG (530B)	64 K→3.75M	5×10^{-5}	yes	cosine decay to 10%	Adam	BF16	0.1	1.0	-
Gopher (280B)	3M→6M	4×10^{-5}	yes	cosine decay to 10%	Adam	BF16	-	1.0	-
Chinchilla (70B)	1.5M→3M	1×10^{-4}	yes	cosine decay to 10%	AdamW	BF16	-	-	-
Galactica (120B)	2M	7×10^{-6}	yes	linear decay to 10%	AdamW	-	0.1	1.0	0.1
LaMDA (137B)	256K	-	-	-	-	BF16	-	-	-
Jurassic-1 (178B)	32 K→3.2M	6×10^{-5}	yes	-	-	-	-	-	-
LLaMA (65B)	4M	1.5×10^{-4}	yes	cosine decay to 10%	AdamW	-	0.1	1.0	-
GLM (130B)	0.4M→8.25M	8×10^{-5}	yes	cosine decay to 10%	AdamW	FP16	0.1	1.0	0.1
T5 (11B)	64K	1×10^{-2}	no	inverse square root	AdaFactor	-	-	-	0.1
ERNIE 3.0 Titan (260B)	-	1×10^{-4}	-	-	Adam	FP16	0.1	1.0	-
PanGu-Σ (1.085T)	0.5M	2×10^{-5}	yes	-	Adam	FP16	-	-	-

以提高训练的稳定性和吞吐量。对于像 GPT-3 和 PaLM 这样的大语言模型，它们引入了一种新的策略，在训练过程中动态增加批量大小，最终达到百万级别。具体而言，GPT-3 的批量大小从 32K 逐渐增加到 3.2M 个标记。实证结果表明，批量大小的动态调整策略可以有效地稳定大语言模型的训练过程 [56]。

学习率： 现有的大语言模型通常在预训练过程中采用类似的学习率调整策略，包括 warm-up 和 decay。具体而言，在训练的初始 0.1% 到 0.5% 的步骤中，采用线性 warm-up 策略逐渐增加学习率到最大值，这个最大值通常在 5×10^{-5} 到 1×10^{-4} 之间（例如 GPT-3 的学习率为 6×10^{-5}）。然后，在后续步骤中采用余弦衰减策略，逐渐将学习率降低到其最大值的约 10%，直到训练损失的收敛。

优化器： Adam 优化器 [175] 和 AdamW 优化器 [176] 被广泛应用于训练大语言模型（例如 GPT-3），这些优化器基于第一阶梯度的自适应估计的低阶矩。通常，它的超参数设置如下：$\beta_1 = 0.9$，$\beta_2 = 0.95$ 和 $\epsilon = 10^{-8}$。同时，Adafactor 优化器 [177] 也被用于训练大语言模型（例如 PaLM 和 T5），它是一种 Adam 优化器的变体，专门设计用于在训练过程中保存 GPU 内存。Adafactor 优化器的超参数设置如下：$\beta_1 = 0.9$，$\beta_2 = 1.0 - k^{-0.8}$，其中 k 表示训练步骤的数量。

稳定训练： 在大语言模型的预训练过程中，常常会遇到训练不稳定的问题，这可能会导致模型崩溃。为了解决这个问题，通常会广泛使用权重衰减和梯度裁剪，其中现有的研究 [55, 68, 80, 82, 97] 通常将梯度裁剪的阈值设置为 1.0，将权重衰减率设置为 0.1。然而，随着大语言模型的扩展，训练损失的峰值也更容易发生，导致训练不稳定。为了缓解这个问题，PaLM [56] 和 OPT [80] 使用了一种简单的策略，即从峰值之前的一个检查点重新开始训练过程，并跳过可能导致问题的数据。此外，GLM [82] 发现嵌入层的异常梯度通常会导致峰值，因此提出缩小嵌入层梯度以缓解这个问题。

4.3.2 可扩展的训练技术

随着模型和数据的规模增加，有限的计算资源下高效地训练大语言模型变得具有挑战性。尤其需要解决两个主要的技术问题，即提高训练吞吐量和将更大的模型加载到 GPU 内存中。在本部分中，我们回顾了现有工作中用于解决上述两个挑战的几种广泛使用的方法，即 3D 并行 [65, 178, 179]，ZeRO [180] 和混合精度训练 [181]，并提供了关于如何利用它们进行训练的一般性建议。

3D 并行： 3D 并行实际上是三种常用并行训练技术的组合，即数据并行、流水线并行 [178, 179] 和张量并行 [65][10]。我们接下来介绍这三种并行训练技术。

- *数据并行*：数据并行是提高训练吞吐量的最基本方法之一。它将模型参数和优化器状态复制到多个 GPU 上，然后将整个训练语料库分配到这些 GPU 上。这样，每个 GPU 只需要处理分配给它的数据，并执行前向和反向传播以获取梯度。在不同 GPU 上计算的梯度将进一步聚合以获得整个批次的梯度，以更新所有 GPU 上的模型。这样，由于梯度的计算在不同 GPU 上是独立进行的，数据并行机制具有高度可扩展性，可以通过增加 GPU 数量来提高训练吞吐量。此外，该技术的实现简单，大多数现有的流行深度学习库已经实现了数据并行，例如 TensorFlow 和 PyTorch。
- *流水线并行*：流水线并行旨在将大语言模型的不同层分布到多个 GPU 上。特别是在 Transformer 模型的情况下，流水线并行将连续的层加载到同一 GPU 上，以减少在 GPU 之间传输计算隐藏状态或梯度的成本。然而，流水线并行的

10. 模型并行是一个更广泛的术语，在一些工作中包括张量并行和流水线并行 [65]。

一种朴素实现可能导致 GPU 利用率降低，因为每个 GPU 必须等待前一个 GPU 完成计算，从而导致不必要的气泡开销 [178]。为了减少流水线并行中的这些气泡，GPipe [178] 和 PipeDream [179] 提出了填充多个数据批次和异步梯度更新技术，以提高流水线效率。

• 张量并行：张量并行也是一种常用的技术，旨在将大语言模型分解为多 GPU 加载。与流水线并行不同，张量并行专注于分解大语言模型的张量（参数矩阵）。对于大语言模型中的矩阵乘法操作 $Y = XA$，参数矩阵 A 可以按列分成两个子矩阵 A_1 和 A_2，可以表示为 $Y = [XA_1, XA_2]$。通过将矩阵 A_1 和 A_2 放置在不同的 GPU 上，矩阵乘法操作将在两个 GPU 上并行调用，并且可以通过跨 GPU 通信将两个 GPU 的输出组合成最终结果。目前，张量并行已经在几个开源库中得到支持，例如 Megatron-LM [65]，并且可以扩展到更高维度的张量。此外，Colossal-AI 还为更高维度的张量实现了张量并行 [182–184]，并提出了序列并行 [185]，特别是针对序列数据，可以进一步分解 Transformer 模型的注意力操作。

ZeRO： ZeRO（Zero Redundancy Optimizer）技术，由 DeepSpeed [64] 库提出，专注于解决数据并行中的内存冗余问题。如前所述，数据并行需要每个 GPU 存储大语言模型的相同副本，包括模型参数、模型梯度和优化器参数。然而，并非所有上述数据都需要在每个 GPU 上保留，这将导致内存冗余问题。为了解决这个问题，ZeRO 技术旨在仅在每个 GPU 上保留部分数据，而当需要时，其余数据可以从其他 GPU 中检索。具体而言，ZeRO 提供了三种解决方案，具体取决于三个数据部分的存储方式，即优化器状态分区、梯度分区和参数分区。实证结果表明，前两种解决方案不会增加通信开销，第三种解决方案会增加约 50% 的通信开销，但可节省与 GPU 数量成比例的内存。PyTorch 实现了与 ZeRO 类似的技术，称为 FSDP [186]。

混合精度训练： 在以前的 PLMs（例如 BERT [23]）中，主要使用 32 位浮点数（FP32）进行预训练。近年来，为了预训练极大的语言模型，一些研究 [181] 开始利用 16 位浮点数（FP16），这可以减少内存使用和通信开销。此外，由于流行的 NVIDIA GPU（例如 A100）具有 FP16 计算单元的两倍，FP16 的计算效率可以进一步提高。然而，现有的研究发现，FP16 可能导致计算精度的损失 [59, 68]，影响最终的模型性能。为了缓解这个问题，一种替代方案称为 *Brain Floating Point (BF16)* 已被用于训练，它比 FP16 分配更多的指数位和更少的有效位。对于预训练，BF16 通常比 FP16 在表示准确性方面表现更好 [68]。

整体训练建议： 在实践中，上述训练技术，特别是三维并行技术，通常会联合使用以提高训练吞吐量和大模型加载。例如，研究人员已经将 8 路数据并行、4 路张量并行和 12 路流水线并行纳入到 BLOOM [68] 的 384 个 A100 GPU 上进行训练。目前，开源库如 DeepSpeed [64]、Colossal-AI [125] 和 Alpa [187] 可以很好地支持这三种并行训练方法。为了减少内存冗余，可以使用 ZeRO、FSDP 和激活计算技术 [67, 188] 来训练大语言模型，这些技术已经集成到 DeepSpeed、PyTorch 和 Megatron-LM 中。此外，混合精度训练技术，如 BF16，也可以利用来提高训练效率和减少 GPU 内存使用，但需要硬件的必要支持（如 A100 GPU）。由于训练大模型是一个耗时的过程，因此在早期阶段预测模型性能并检测异常问题将非常有用。为此，GPT-4 [45] 最近引入了一种基于深度学习堆栈的新机制，称为可预测扩展，可以使用更小的模型对大模型进行性能预测，这对于开发大语言模型可能非常有用。在实践中，人们还可以进一步利用主流深度学习框架的支持训练技术。例如，PyTorch 支持完全分片数据并行训练算法 FSDP [186]，如果需要，可以将部分训练计算卸载到 CPU 上。

除了上述训练策略，提高使用大语言模型的推理速度也很重要。通常，量化技术被广泛用于在推理阶段减少大语言模型的时间和空间成本 [189]。虽然会损失一些模型性能，但量化语言模型具有更小的模型大小和更快的推理速度 [82, 190, 191]。对于模型量化，INT8 量化是一个流行的选择 [190]。此外，一些研究工作尝试开发更激进的 INT4 量化方法 [82]。在这些开源大语言模型中，BLOOM[11]、GPT-J[12]和 GLM[13]已经发布了相应的量化模型副本。

5 大语言模型的适配微调

在预训练后，大语言模型可以获得解决各种任务的通用能力。然而，越来越多的研究表明，大语言模型的能力可以根据特定目标进一步调整。本节中，我们介绍了两种微调预训练后的大语言模型的方法：指令微调和对齐微调。前一种方法旨在增强（或解锁）大语言模型的能力，而后一种方法旨在将大语言模型的行为与人类的价值观或偏好对齐。接下来，我们将详细介绍这两种方法。

表 5
可用于指令微调的任务集合的详细列表。注意，OIG 是一个包含现有集合的大型集合。

Collections	Time	#Task types	#Tasks	#Examples
Nat. Inst. [192]	Apr-2021	6	61	193K
CrossFit [193]	Apr-2021	13	160	7.1M
FLAN [62]	Sep-2021	12	62	4.4M
P3 [194]	Oct-2021	13	267	12.1M
ExMix [195]	Nov-2021	11	107	18M
UnifiedSKG [196]	Jan-2022	6	21	812K
Super Nat. Inst. [78]	Apr-2022	76	1616	5M
MVPCorpus [197]	Jun-2022	11	77	41M
xP3 [84]	Nov-2022	17	85	81M
OIG[14]	Mar-2023	-	-	43M

11. https://huggingface.co/joaoalvarenga/bloom-8bit
12. https://huggingface.co/hivemind/gpt-j-6B-8bit
13. https://github.com/ggerganov/llama.cpp
14. https://laion.ai/blog/oig-dataset/

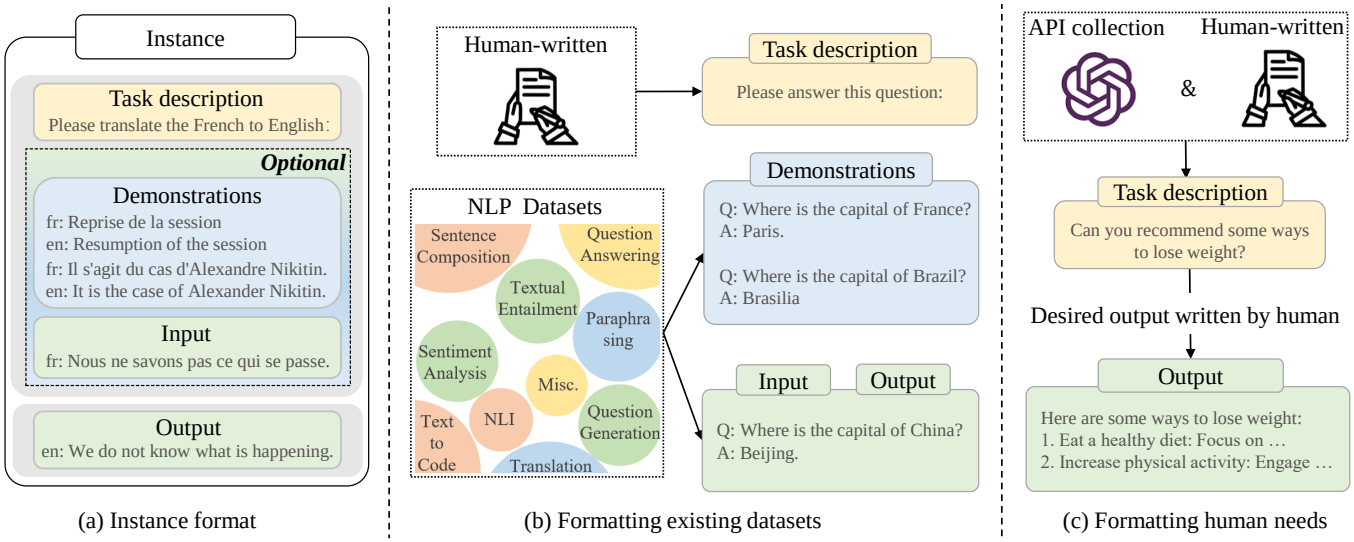


图 4. 实例格式化和两种构造指令格式实例的示意图。

5.1 指令微调

本质上，指令微调是在自然语言格式的实例集合上微调预训练后的大语言模型的方法 [62]。这种方法与有监督微调 [61] 和多任务提示训练 [28] 密切相关。为了进行指令微调，我们首先需要收集或构造指令格式的实例。然后，我们使用这些格式化的实例以有监督的方式微调大语言模型（例如，使用序列到序列的损失进行训练）。指令微调后，大语言模型展现出泛化到未见过任务的卓越能力 [28, 62, 83]，即使在多语言场景下也能有不错表现 [84]。

最近的一篇综述 [198] 对指令微调进行了系统的概述。相比之下，我们主要关注指令微调对大语言模型的影响，并提供了收集实例和微调模型的详细策略或指南。此外，我们还讨论了使用指令微调来满足用户的实际需求，这在现有的大语言模型中被广泛应用，例如 InstructGPT [61] 和 GPT-4 [45]。

5.1.1 格式化实例构造

通常情况下，一个指令格式化的实例包括一个任务描述（称为指令）、一个输入输出对以及少量示例（可选）。作为重要的公共资源，现有的研究已经发布了大量标注为自然语言格式的数据（可见表5 中的可用资源列表）。接下来，我们将介绍构建格式化实例的两种主要方法（可见图4 中的插图），然后讨论实例构建的几个关键因素。

格式化已有数据集：在指令微调被提出之前，几项早期的研究 [195, 197, 199, 200] 通过收集来自不同领域（例如文本摘要、文本分类和翻译）的实例来创建有监督的多任务训练数据集。作为指令微调实例的重要来源，使用自然语言的任务描述格式化这些多任务训练数据集是相当方便的。具体来说，最近的工作 [28, 61, 62, 78] 使用人类编写的任务描述来增强标记的数据集，这些描述通过解释任务目标来指导 LLM 理解任务。例如，在图 4(b) 中，针对问答任务中的每个实例都添加了一个任务描述“请回答下列问题”。在指令调整之后，LLM 可以通过遵循它们的任务描述很好地泛化到其他看不见的任务 [28, 62, 83]。特别的，指令被证明是影响 LLM 任务泛化能力的关键因素 [62]。为了更好地为指令微调生成标注实例，一种名为 PromptSource 的众包平台 [194] 被提出，可以有效地创建、共享和验证不同数据集的任务描述。此外，一些研究 [28, 197, 201] 还尝试通过反转现有实例的输入-输出对，并针对指令微调设计专门的任务描述。例如，给定一个问题-答案对，我们可以通过预测以问题和一些任务描述（例如，“请基于以下答案按生成一个问题：”）为条件的答案，来创建一个新实例。此外，一些工作 [202] 还利用启发式任务模板将大量无标注文本转换为标注实例。

格式化人类需求：尽管大量的训练实例已经通过指令进行了格式化，但它们主要来自于公共 NLP 数据集，任务描述缺乏多样性或与真正的人类需求不匹配 [61]。为了解决这个问题，InstructGPT [61] 建议采用真实用户提交给 OpenAI API 的查询作为任务描述。用户查询以自然语言表示，很适合引出 LLM 遵循指令的能力。此外，为了丰富任务的多样性，还要求人工标注者为真实生活中的任务编写指令，包括开放式生成、开放式问答、头脑风暴和聊天等。然后，他们让另一组标注人员直接按照这些指令作为输出进行回答。最后，将一个指令（即采集的用户查询）和期望的输出（即人工编写的答案）配对作为一个训练实例。值得注意的是，InstructGPT 还将这些以自然语言格式化的真实世界任务用于对齐调整（在第 5.2 节中讨论）。进一步，GPT-4 [45] 还设计了具有潜在高风险的指令，并通过监督微调指导模型拒绝这些指令以确保安全。除了这些之外，为了减轻人工注释的负担，几种半自动化的方法 [203–205] 被提出，通过将现有实例输入到 LLM 中生成多样的任务描述和实例来构建实例。

实例构建的关键因素。指令实例的质量对模型的性能有重要的影响。在此，我们讨论了一些实例构建中的关键因素。

• *扩展指令*：大量研究已经证明扩大任务数量可以极大地提高 LLM 的泛化能力 [28, 62, 78]。随着任务数量的增加，模型性能最初呈现连续增长的趋势，但当达到一定水平时，对模型性能的提升变得微不足道 [78, 83]。一个合理的猜测是，一定数量的代表性任务可以提供相对充足的知识，添加更多的任务可能不会带来额外的收益 [83]。此外，从多个方面增强任务描述的多样性也是有益的，例如长度、结构和创造力 [28]。至于每个任务所需的实例数量，已有研究发现少量实例通常可以使模型的泛化性能达到饱和 [62, 83]。然而，将某些任务的实例数量进一步增加（例如数百个）可能会潜在地导致过拟合问题并影响模型性能 [78]。

• *格式设计*：指令的格式设计也是影响 LLM 泛化性能的一个重要因素 [78]。通常，我们可以向现有数据集的输入-输出对添加任务描述和可选的示例，其中任务描述是 LLMM 理解任务 [78] 的最关键部分。此外，通过使用适当数量的示例作为演示 [83]，可以导致实质性的改进，这也减轻了模型对指令工程的敏感性 [62, 83]。然而，将其他组件 (*e.g.,* 要避免的事情、原因和建议) 合并到指令中可能会对 LLMs [78, 192] 的性能提升产生很轻微的甚至不利的影响。最近，为了引出 LLM 的逐步推理能力，一些工作 [83] 建议包含一些面向推理数据集的思维 (CoT) 示例，例如算术推理。已经有研究表明，同时使用包含 CoT 和非 CoT 样本微调 LLM，可以在各种下游任务中获得良好的性能，包括那些需要多跳推理能力的任务（例如，常识问答和算术推理），以及那些不需要这种推理能力的任务（例如，情感分析和抽取式问答） [83, 85]。

总的来说，指令的多样性看起来比实例数量更重要，因为表现良好的 InstructGPT [61] 和 Alpaca [205] 使用的指令（或实例）比 Flan 系列 LLM [62, 83] 更少但更加多样化。此外，邀请标注者构建人类需求任务比使用特定于数据集的任务更有用。但是，仍然缺乏如何标注满足人类需求指令的指南，使任务构建多少具有一定的启发性。为了减少人力成本，我们可以重用现有的格式化数据集（表5），或使用现有的 LLM 自动构建指令 [203]。

5.1.2 指令微调策略

与预训练不同，指令微调通常更加高效，其只需要使用一定数量的实例进行训练。指令微调可以被视为一个有监督的训练过程，其优化过程与预训练有一些不同 [83]，比如训练目标（如序列到序列损失）和优化配置（如更小的批量大小和学习率），这些细节需要在实践中特别注意。除了这些优化配置，指令微调还需要考虑两个重要方面：

平衡数据分布：由于指令微调涉及多种任务的混合，因此在微调过程中平衡不同任务的比例非常重要。一种广泛使用的方法是*实例比例混合策略* [72]，即将所有数据集合并，然后从混合数据集中等比例采样每个实例。此外，根据最近的研究发现 [83, 85]，提高高质量数据集（例如 FLAN [62] 和 P3 [194]）的采样比例通常可以带来性能提升。同时，在指令微调期间通常会设置一个*最大容量*，以限制数据集中可以包含的最大实例数 [72]，这是为了防止较大的数据集压倒整个分布 [72, 85]。在实践中，根据不同的数据集，最大容量通常设置为几千或几万个实例 [62, 83]。

结合指令微调和预训练：为了使微调过程更加有效和稳定，OPT-IML [85] 在指令微调期间加入了预训练数据，这可以看作是对模型的正则化。此外，一些研究并没有使用单独的两阶段训练过程（*预训练然后指令微调*），而是尝试使用多任务学习从头开始训练模型，混合使用预训练数据（即纯文本）和指令微调数据（即指令格式数据） [72, 195]。具体而言，GLM-130B [82] 和 Galactica [34] 将指令格式数据集作为预训练语料库的一小部分来预训练大语言模型，这有可能同时获得预训练和指令微调的优势。

5.1.3 指令微调的效果

在这部分中，我们讨论了指令微调对大语言模型的两个主要方面的影响。

性能改进：尽管指令微调仅在有限数量的数据上进行了微调，但已成为改进或发掘大语言模型能力的重要方式 [83]。最近的研究在多个规模（从 77M 到 540B 不等）上对语言模型进行了实验，表明不同规模的模型都可以从指令微调中受益 [83, 201]，随着参数规模的增加，性能也得到了提升 [84]。此外，经过指令微调的较小模型甚至可以比未经微调的较大模型表现更好 [28, 83]。除了模型规模外，指令微调在不同的模型架构、预训练目标和模型适应方法上都展现出持续的改进效果 [83]。在实践中，指令微调为提升现有语言模型（包括小型预训练语言模型）的能力提供了一种通用的方法 [83]。此外，与预训练相比，指令微调成本较低，因为大语言模型所需的指令数据数量明显较少于预训练数据。

任务泛化性：指令微调鼓励模型理解用于任务完成的自然语言指令。它赋予大语言模型遵循人类指令执行特定任务的能力（通常被视为一种涌现能力），即使在未见过的任务上也能够执行 [83]。大量研究已经证实了指令微调在已见和未见任务上实现卓越的性能表现 [85, 201]。此外，指令微调还被证明对缓解大语言模型的一些弱点（如生成重复内容或在不完成特定任务的情况下补充输入）具有帮助 [61, 83]，从而使大语言模型具有更强的解决现实世界任务的能力。此外，通过使用指令微调训练的大语言模型可以在不同语言之间泛化到相关任务。例如，BLOOMZ-P3 [84] 基于 BLOOM [68] 进行微调，使用仅包含英语的 P3 任务集合 [194]。有趣的是，与 BLOOM 相比，BLOOMZ-P3 在多语言句子完成任务中可以实现超过 50% 的性能提升，这表明指令微调可以帮助大语言模型从仅包含英语的数据集中获取一般的任务技能，并将这些技能转移到其他语言 [84]。此外，研究还发现，在多语言任

务中，使用仅包含英语的指令可以产生令人满意的结果 [84]，从而减少了针对特定语言的指令工程的工作量。

5.2 对齐微调

这部分首先介绍语言模型对齐的背景，包括其定义和标准，然后重点讨论收集人类反馈数据以对齐大语言模型（LLM），最后探讨从人类反馈中进行强化学习以进行语言模型对齐的关键技术。

5.2.1 对齐微调的背景

背景：大语言模型在多个自然语言处理任务上展示出了惊人的能力 [55, 56, 62, 80]。但是，这些模型有时可能表现出意外的行为，例如制造虚假信息、追求不准确的目标，以及产生有害的、误导性的和偏见性的表达 [61, 206]. 对于 LLM，语言建模目标通过单词预测对模型参数进行预训练，但缺乏对人类价值观或偏好的考虑。为了避免这些意外行为，研究提出了人类对齐，使大语言模型行为能够符合人类的期望 [61, 100]。但是，与初始的预训练和适应微调（例如指令微调）不同，语言模型的对齐需要考虑不同的标准（例如有用性，诚实性和无害性）。已有研究表明对齐微调可能会在某种程度上损害大语言模型的通用能力，这在相关研究中被称为对齐税 [61, 207, 208]。

对齐标准：近期，越来越多的研究致力于为规范大语言模型的行为制定多样化的标准。在此，我们选取三个具有代表性的对齐标准（即有用性、诚实性、无害性）作为讨论实例，这些标准已在现有文献中得到广泛采纳 [61, 206, 207]。除此以外，从不同视角出发，还有其他针对大语言模型的对齐标准，涵盖了行为、意图、激励和内在层面 [206]，这与上述三个对齐标准在本质上是相似的，或者使用了相似的对齐技术。根据不同需求，修改上述三个对齐标准也是可行的，例如将诚实性替换为正确性 [100]，或者关注某些特定的标准 [208]。接下来，我们将对上述三个代表性的对齐标准给出简要的解释：

- 有用性：为了具有帮助性，大语言模型应当尽其所能以简明扼要且高效的方式帮助用户解决任务或回答问题。在更高层次上，当需要进一步澄清时，大语言模型应展示出通过相关提问获取额外相关信息的能力，并表现出合适的敏感度、洞察力和审慎程度 [207]。实现大语言模型有用行为的对齐具有挑战性，因为准确定义和衡量用户意图是困难的 [206]。
- 诚实性：在基本层面上，诚实的大语言模型应该向用户提供准确的内容，而不会捏造信息。此外，大语言模型在输出时传达适当程度的不确定性至关重要，以避免任何形式的欺骗或信息误传。这需要模型了解其能力和知识水平（例如“知道未知”）。根据过去的研究 [207]，与有用性和无害性相比，诚实性是一个更客观的标准，因此在诚实性上的对齐可能较少依赖人力。
- 无害性：要做到无害，这要求模型生成的语言不得具有冒犯性或歧视性。在最大限度地发挥其能力的前提下，模型应能够检测到旨在搜集恶意目的请求的隐蔽行动。理想情况下，当模型被诱导去执行危险行为（如犯罪）时，大语言模型应礼貌地拒绝。然而，哪些行为被认为是有害的以及多大程度的有害因个人或社会的差异而不同 [207]，这在很大程度上取决于谁在使用大语言模型、提出的问题类型以及使用大语言模型的背景（如时间）。

正如我们所看到的，这些对齐标准相当主观，并基于人类认知进行制定。因此，将它们直接制定为大语言模型的优化目标是困难的。在现有的研究中，有许多方法可以在对齐大语言模型时满足这些标准。一个有前景的技术是红队攻防 [209, 210]，它涉及使用手动或自动手段以对抗性方式探测大语言模型，生成有害输出，然后更新大语言模型以防止此类输出。

5.2.2 人类反馈收集

在预训练阶段，大语言模型使用大规模语料库，以语言建模为训练目标进行训练。然而，这样的训练目标缺乏人类对大语言模型输出的主观和定性评估（在本综述中称为人类反馈）。高质量的人类反馈对于调整大语言模型与人类偏好和价值观的一致性非常重要。在本部分，我们将讨论如何筛选出优秀的人类标注者来进行反馈数据收集。

标注人员选择：现有的工作中，生成人类反馈数据的主要方法是人工标注 [61, 100, 211]。这凸显了人类标注者在反馈收集的重要作用。为了提供高质量的反馈数据，标注人员应具有合格的教育水平和优秀的英语能力。例如，Sparrow [100] 要求标注人员在英国出生，母语为英语，并且至少获得本科学历。此外，在 [208] 中，高优先级任务中的约一半标注人员是从拥有硕士学位的 Amazon Mechanical Turk 工作人员中招募的。即便如此，一些研究 [211, 212] 发现标注人员与研究人员的意图仍然会存在不匹配的问题，这可能导致低质量的人类反馈并导致大语言模型产生意外的输出。为了解决这个问题，InstructGPT [61] 进一步进行筛选过程，通过评估标注人员与研究人员之间意图的一致性来筛选标注人员。具体而言，研究人员首先进行少量的数据的标注，然后衡量他们自己和标注人员之间的一致性。选择一致性最高的标记者继续进行后续的注释工作。在一些其他的工作中 [213]，使用“优秀标注者”来确保人类反馈的高质量。研究人员评估标注人员的表现，并选择一组表现良好的人类标注人员（例如高一致性）作为优秀标注者。优秀标注者将优先与研究人员合作进行后续的研究。在标注人员进行标注的过程中，提供详细的标注指令与实时的指导是有帮助的 [212]，这可以对标注结果进一步的规范化。

人类反馈收集：在现有的工作中，主要有三种方法从人类标注者中收集反馈和偏好数据。

- 基于排序的方法：在早期的工作中 [211, 214]，标注人员通常以较为粗略的方式（从若干个候选中选择最佳的）评估模型生成的输出，而不考虑更精细的对齐标准。然而，不同

的标注者可能对最佳候选结果的选择持有不同的意见，同时，这种方法忽略了未被选中的样本，可能导致不准确或不完整的人类反馈。为了解决这个问题，随后的研究 [100, 208] 引入了 Elo 评分系统，通过对候选生成结果进行一一比较来得到一个候选结果的排序。输出的排序结果将用于指导模型偏好某些输出而不是其他输出，从而产生更可靠和更安全的输出。

● 基于问题的方法：此外，标注人员可以通过回答研究人员设计的特定问题来提供更详细的反馈 [71]，这些问题能够涵盖不同的对齐标准以及对大语言模型的其他约束条件。特别地，在 WebGPT [71] 中，为了帮助模型从检索到的文档中过滤和利用相关信息，标注人员需要回答关于检索到的文档是否有用于回答给定输入的问题，并提供多个选项。

● 基于规则的方法：此外，在许多研究中，基于规则的方法也被用来提供更详细的人类反馈。作为一个经典模型，Sparrow [100] 不仅选择要求标注人员挑选最佳的回复，还设计了一系列规则来测试模型生成的回复是否符合有用、正确和无害的对齐标准。通过这种方式，研究人员可以获得两种类型的人类反馈数据：(1) 通过比较模型生成的输出对的质量来获得单纯的偏好反馈，和 (2) 通过收集人类标注者的评估（即表示生成的输出违反规则的程度）来获得规则反馈。此外，GPT-4 [45] 利用一组零样本分类器（基于 GPT-4 本身）作为基于规则的奖励模型，可以自动确定模型生成的输出是否违反了一组人类编写的规则。

接下来，我们将重点关注一种被广泛应用于大语言模型（如 ChatGPT）中的技术，即基于人类反馈的强化学习（RLHF）。在下面的章节里，我们将介绍如何通过让大语言模型在用户请求中学习人类反馈来实现在第5.2.1节中介绍到的对齐标准。

5.2.3 基于人类反馈的强化学习

为了使 LLM 与人类价值观保持一致，人们提出了基于人类反馈的强化学习（RLHF）[69, 211]，使用收集到的人类反馈数据对 LLM 进行微调，有助于改进对齐标准（例如，有用性，诚实性和无害性）。RLHF 采用强化学习（RL）算法（例如，近端策略优化（PPO）[215]）通过学习奖励模型使 LLM 适应人类反馈。这种方法将人类纳入训练的循环中来开发对齐性能良好的 LLM，如 InstructGPT [61].

RLHF 系统。RLHF 系统主要包括三个关键组件：要对齐的预训练 LM、从人类反馈中学习的奖励模型以及训练 LM 的 RL 算法。具体来说，预训练 *LM* 通常是一个生成模型，它使用现有的预训练 LM 参数进行初始化。例如，OpenAI 在其第一个主流的 RLHF 模型 InstructGPT [61] 中使用 175B 参数量的 GPT-3，而 DeepMind 在其 GopherCite 模型 [213] 中使用 2800 亿参数模型 Gopher [59]. 此外，奖励模型（RM）提供（学习的）指导信号，这些信号反映了人类对 LM 生成的文本的偏好，通常以标量值的形式存在。奖励模型可以采用两种形式：微调的 LM 或使用人类偏好数据重新训练的 LM。现

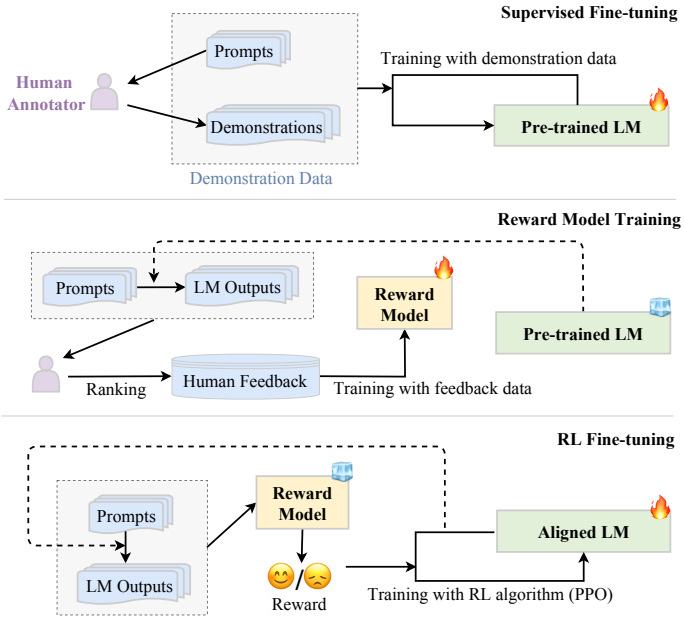


图 5. RLHF 算法工作流。

有工作通常采用具有与需要对齐的 LM [61, 213] 不同的参数尺度的奖励模型。例如，OpenAI 使用 6B 参数量的 GPT-3，DeepMind 使用 7B 参数量的 Gopher 作为奖励模型。最后，为了使用来自奖励模型的信号优化预训练 LM，设计了一种特定的 RL 算法用于大规模模型调整。具体地，近端策略优化（Proximal Policy Optimization, PPO）[215] 是一种在现有工作中广泛使用的 RL 对齐算法 [61, 100, 213].

RLHF 的关键步骤。图 5说明了 RLHF 的整个三步过程 [61, 212]，具体如下所述。

● 监督微调。为了使 LM 具有初步执行所需行为的能力，通常需要收集一个包含输入提示（指令）和所需输出的监督数据集，以对 LM 进行微调。这些提示和输出可以由人工标注人员针对某些特定任务编写，同时确保任务的多样性。例如，InstructGPT [61] 要求人工标注者编写提示（例如，“List five ideas for how to regain enthusiasm for my career”）和一些生成式任务（如开放域问答、头脑风暴、聊天和重写）的期望输出。请注意，第一步在特定设置或场景中是可选的。

● 训练奖励模型。第二步是使用人类反馈数据训练 RM。具体来说，使用 LM 使用采样提示（来自监督数据集或人工生成的提示）作为输入来生成一定数量的输出文本，然后邀请人工标注员为这些对标注偏好。标注过程可以以多种形式进行，常见的做法是对生成的候选文本进行排序标注，这样可以减少标注者之间的不一致性。然后，需要训练 RM 预测人类偏好的输出。在 InstructGPT 中，标注员将模型生成的输出从最好到最差进行排名，然后训练 RM（即 6B 参数量的 GPT-3）来预测排名。

● *RL* 微调. 在这一步骤中，对齐（即微调）LM 被形式化为 RL 问题。在此设置中，预训练的 LM 作为策略，将提示

作为输入并返回输出文本，它的动作空间是 LM 的词表，状态是当前生成的 token 序列，奖励由 RM。为了避免显着偏离初始（调整前）LM，通常将惩罚项纳入奖励函数。例如，InstructGPT 使用 PPO 算法针对 RM 优化 LM。对于每个输入提示，InstructGPT 计算当前 LM 和初始 LM 生成的结果之间的 KL 散度作为惩罚。值得注意的是，第二步和最后一步可以多次迭代来更好地对齐 LLM。

6 使用

经过预训练或适应微调之后，使用大语言模型的主要方法是为解决各种任务设计适当的提示策略。典型的提示方法是上下文学习 [50, 55]，它将任务描述和/或样例以自然语言文本的形式表达。此外，可以通过将一系列中间推理步骤纳入提示中，采用思维链提示 [32] 来增强上下文学习。接下来，我们将详细介绍这两种技术的细节。

6.1 上下文学习

作为一种特殊的提示形式，“上下文学习”（ICL）首次在 GPT-3 [55] 中被提出，并成为利用大语言模型的典型方法。

6.1.1 上下文学习的一般形式

根据 Brown 等人给出的定义 [55]，ICL 使用一种由任务描述和/或几个任务样例作为示范组成的自然语言提示。图6展示了 ICL 的示意图。首先，从任务数据集中选择一些样例作为示范。然后，按照特定的顺序将它们与特别设计的模板组合成自然语言提示。最后，将测试样例添加到大语言模型的输入中以生成输出。基于任务示范，大语言模型可以在没有显式梯度更新的情况下识别和执行新任务。

正式地，设 $D_k = \{f(x_1, y_1), \ldots, f(x_k, y_k)\}$ 代表由 k 个样例组成的一组示范，其中 $f(x_k, y_k)$ 是将第 k 个任务样例转换为自然语言提示的函数。给定任务描述 I、样例 D_k 以及新的输入查询 x_{k+1}，大语言模型生成的输出 $\hat{y}_{k+1}$ 的预测可以用如下公式表示[15]:

$$\mathrm{LLM}\big(I, \underbrace{f(x_1, y_1), \ldots, f(x_k, y_k)}_{\text{样例}}, f(\underbrace{x_{k+1}}_{\text{输入}}, \underbrace{\_\_\_\_}_{\text{答案}})\big) \rightarrow \hat{y}_{k+1}. \tag{3}$$

答案 y_{k+1} 由大语言模型的预测结果给出。因为 ICL 的性能主要取决于样例，如何在提示中合理地设计它们是一个重要的问题。根据式 (3)的构建过程，我们着重于提示中示范的设计，包括如何选择组成示范的样例，用函数 $f(\cdot)$ 将每个样例格式化到提示中，以及如何合理地排列这些样例。

调研论文 [50] 对 ICL 进行了全面的综述，建议读者参考它以获得关于此主题的更加普遍和详细的讨论。相比之下，我们特别关注两个方面，即样例设计以及 *ICL* 背后的机制。此外，ICL 还与指令微调（在5.1中讨论）有着密切的联系，因为它们都将任务或样例转化为自然语言的形式。但是，指令微调需要微调大语言模型的权重，而 ICL 不涉及微调，仅仅是使用大语言模型的一种方式。此外，指令微调可以提高大语言模型执行目标任务的 ICL 能力，尤其是在零样本设置下（仅使用任务描述）[83]。

15. ICL 在 GPT-3 论文 [55] 中首次被介绍时，它被定义为任务描述和示范样例的组合，其中任何一个部分都是可选的。按照这个定义，当大语言模型仅通过任务描述来解决一个未见过的任务时，也可以被视为通过 ICL 来解决任务，指令微调可以增强这种情况下的 ICL 能力。

6.1.2 样例设计

多项研究表明，ICL 的有效性在很大程度上受到样例设计的影响 [216–218]。根据6.1.1节中的讨论，我们将主要从三个方面介绍 ICL 的样例设计，即样例选择、格式和顺序。

样例选择。 根据 [219]，不同的样例对于 ICL 的性能影响非常大。因此，选择一个能够有效发挥大语言模型 ICL 能力的样例子集是非常重要的。关于样例选择的主要方法有两种，即启发式方法和基于大语言模型的方法：

- *启发式的方法*。由于其简单性和低成本，现有工作广泛采用启发式方法来选择样例。一些研究采用基于 k-NN 的检索器来选择与查询语义相关的样例 [219, 220]。然而，他们只是针对每一个样例进行单独选择，而不是对整个样例集合进行评估。为了解决这个问题，一些研究提出了基于多样性的选择策略来选择特定任务的最具代表性的样例子集 [221, 222]。此外， [223] 在选择样例时同时考虑了相关性和多样性。
- *基于大语言模型的方法*。另一部分工作利用大语言模型来选择示例。例如，大语言模型可以直接根据添加样例后的性能提升 [224] 评估每个样例的信息量以进行选择。此外，EPR [225] 提出了一种两阶段检索方法，首先使用无监督方法（例如 BM25）召回类似的示例，然后使用密集检索器（使用大语言模型标记的正负样例训练）对它们进行排名。样例选择的任务还可以建模为一个强化学习问题，其中大语言模型作为奖励函数，为训练策略模型提供反馈 [226]。因为大语言模型在文本标注方面表现良好 [227]，一些研究采用大语言模型本身作为样例生成器 [228, 229]。

总而言之，正如 [230] 中所讨论的，针对上述两种选择方法，ICL 中所选择的样例应该包含足够的有关要解决任务的信息，并且与测试查询相关。

样例格式。在选择任务样例之后，下一步是将它们格式化为大语言模型的自然语言提示。一种直接的方法是用相应的输入输出对来实例化预定义的模板 [35]。为了构建更具信息量的模板，一些研究 [83] 考虑添加任务描述，或者通过 CoT 提示 [32] 来增强大语言模型的推理能力。例如，在 [192] 中，作者收集了一个由人类编写的大规模指令数据集。大语言模型使用这个数据集微调后，可以提升已见任务的性能，还可以在一定程度上泛化到未见任务上。为了降低注释成本，在 [203] 中作者提出了一种半自动化方法，该方法通过使用由人工编写的任务描述组成的种子集来提示大语言模型生成新任务的任务描

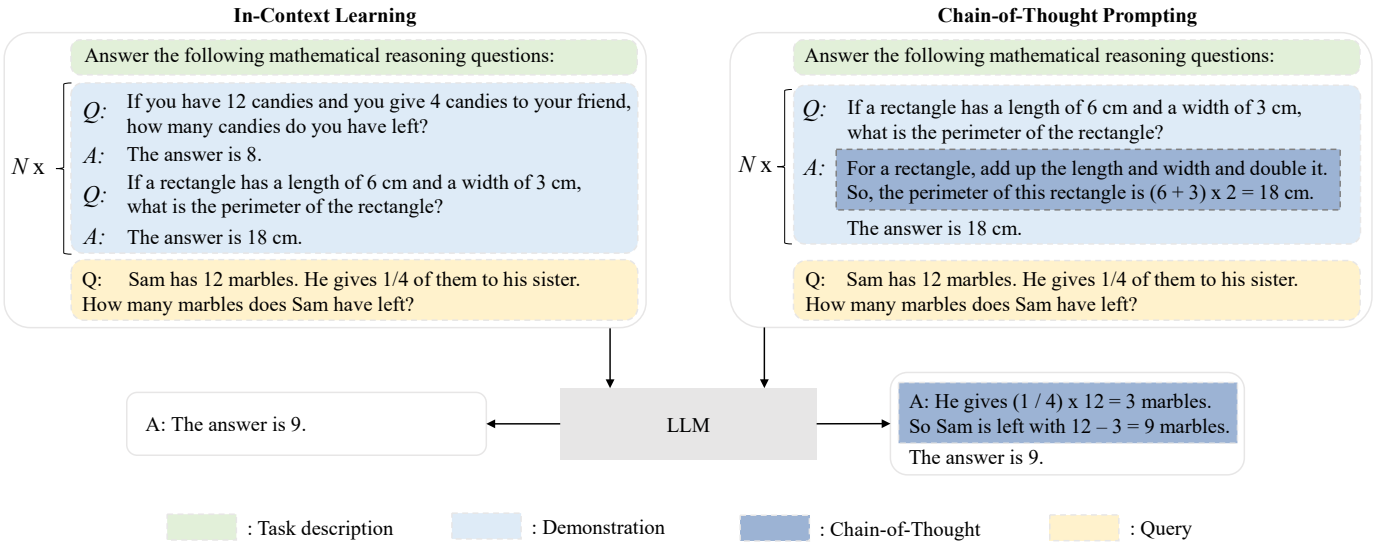


图 6. 一个关于上下文学习（ICL）和思维链（CoT）提示的比较说明。ICL 通过自然语言描述、多个样例和查询来提示大语言模型。而 CoT 提示涉及提示中的一系列中间推理步骤。

述。由于人工标注不同任务的样例格式成本较高，一些工作研究了如何自动生成高质量的样例格式。作为两种典型的方法，Auto-CoT [231] 利用大语言模型，使用零样本提示 "*Let's think step by step*" 来生成中间推理步骤，而 least-to-most 提示 [232] 首先查询大语言模型进行问题分解，然后按顺序依次解决他们，其中的每个子问题都在提示中添加了之前解决的子问题的答案。

样例顺序。大语言模型有时会受到顺序偏差的影响，即倾向于重复样例结尾处的答案 [218]。因此，以合理的顺序排列样例（即任务示例）非常重要。早期的工作提出了一些启发式方法来快速地找到一个良好的顺序。例如，可以直接根据嵌入空间中与查询的相似度排列样例 [219]：相似度越高，距离结尾越近。此外，全局和局部熵度量可以用来给不同的样例顺序打分 [217]。受到信息论的启发，近期研究提出最小化压缩和传输任务标签所需的码长来整合更多的任务信息 [233]。然而，这些方法需要额外的标记数据作为验证集来评估性能。为了消除标注验证集的需要，在 [217] 中作者提出从大语言模型本身采样数据作为验证集。

6.1.3 底层机制

经过预训练，大语言模型可以在不更新梯度的情况下表现出令人惊艳的 ICL 能力。在接下来的内容中，我们将讨论关于大语言模型 ICL 能力的两个关键问题，即 "*预训练如何影响 ICL 能力*" 和 "*大语言模型如何在推理阶段执行 ICL*"。

预训练如何影响 ICL? ICL 首次在 GPT-3 [55] 中提出，其表明 ICL 的能力随着模型尺寸的增大而增强。而有些研究表明，小规模的预训练语言模型也可以通过特别设计的训练任务来展示出强大的 ICL 能力（例如学习根据由任务实例和查询组成的输入来预测标签），甚至可能超越规模更大的模型 [234]。因此，训练任务的设计是影响大语言模型的 ICL 能力的一个重要因素。除了训练任务之外，近期研究还探索了 ICL 与预训练语料之间的关系 [230, 235, 236]。研究表明，ICL 的性能主要取决于预训练语料的来源而非规模 [236]。另一项研究 [235] 深入分析了训练数据分布的影响。他们发现，当训练数据可以被聚类成许多不常见的类别，而不是均匀分布，模型就会出现 ICL 的能力。此外，在 [230] 中作者们从理论上给出了解释，他们认为 ICL 是在具备长程连贯性的文档上进行预训练的产物。

大语言模型如何实现 ICL? 在推理阶段，因为 ICL 不涉及显式的学习或更新，研究人员侧重于分析 ICL 能力和给定的样例之间的关系。他们通常从梯度下降的角度进行分析，并将 ICL 视为隐式微调 [60, 237]。根据这一框架，ICL 可以解释为：通过前向计算，大语言模型生成关于样例的元梯度，并通过注意力机制隐式地执行梯度下降。实验也表明，大语言模型中的某些注意力头能够执行与 ICL 能力密切相关的任务无关的原子操作（例如复制和前缀匹配） [238, 239]。为了进一步探索 ICL 的工作机制，一些研究将 ICL 抽象为一种算法学习过程 [240–242]。具体来说，在 [241] 中作者发现，在预训练阶段大语言模型在参数中隐式地编码了一个模型，在推理阶段，通过 ICL 中提供的示例，大语言模型可以实现诸如梯度下降之类的学习算法，或者直接计算出闭式解以更新这些模型。基于这一框架，研究人员发现大语言模型能够有效地学习简单的线性函数，甚至一些复杂的函数如决策树 [240–242]。

6.2 思维链提示

思维链（Chain-of-Thought, CoT）[32] 是一种改进的提示策略，旨在提高大语言模型在复杂推理任务中的性能，例如算术推理 [243–245]，常识推理 [246, 247] 和符号推理 [32]。CoT

不同于 ICL 仅通过输入-输出对构建提示，而是将可以导出最终输出的中间推理步骤纳入提示中。下面我们将详细介绍使用 CoT 进行 ICL 的方法，并讨论 CoT 提示何时以及为何起作用。

6.2.1 用 CoT 进行上下文学习

通常情况下，CoT 可以在少样本和零样本设置这两种设置下与 ICL 一起使用。

少样本 CoT。少样本 CoT 是 ICL 的一个特例，它通过加入 CoT 推理步骤将每个示范 ⟨ 输入, 输出 ⟩ 扩充为 ⟨ 输入, *CoT*, 输出 ⟩。我们接下来将讨论使用 CoT 的两个关键问题，即如何设计合适的 CoT 提示以及如何利用生成的 CoT 推导出最终答案。

• *CoT* 提示设计。设计合适的 CoT 提示对于有效引出大语言模型的复杂推理能力至关重要。一种直接的方法是使用多样的 CoT 推理路径（即每个问题的多个推理路径），这可以有效增强性能 [248]。另一个直觉的想法是，具有复杂推理路径的提示更有可能引出大语言模型的推理能力 [249]。然而，这两种方法都需要标注 CoT，这限制了它们在实践中的应用。为了克服这个限制，Auto-CoT [231] 提出了利用 Zero-shot-CoT [250]（详见零样本 CoT 部分）让大语言模型生成推理路径。为了提高性能，Auto-CoT 进一步将训练集中的问题分成不同的簇，并选择最接近每个簇质心的问题，它们可以很好地代表整个训练集。尽管少样本 CoT 被视为 ICL 的一种特殊提示情况，但相比于 ICL 中的标准提示，示范的顺序似乎对性能影响相对较小：在大多数任务中，重新排序样例导致的性能变化少于 2% [32]。

• 增强的 *CoT* 策略。除了丰富上下文信息外，CoT 提示还提供了更多推断答案的选项。现有研究主要关注如何生成多个推理路径，并尝试在对应的答案中寻求共识 [251–253]。例如，*self-consistency*[251] 首先用大语言模型生成多个推理路径，然后对所有答案进行集成（例如通过在这些路径之间进行投票来选择最一致的答案）。Self-consistency 极大地提高了 CoT 推理的性能，甚至可以改善一些效果差于标准提示的任务（例如闭卷问答和自然语言推理）。此外，[252] 中的作者将 self-consistency 策略扩展为更通用的集成框架 （提示的集成），他们发现多样化的推理路径是 CoT 推理性能提高的关键。上述方法可以很容易地集成到 CoT 提示中以提高性能，而无需进行额外的训练。其他研究则通过训打分模型来衡量生成的推理路径的可靠性 [248]，或者持续地使用大语言模型自己生成的推理路径进行训练 [254, 255] 以提高性能。

零样本 CoT。与少样本 CoT 不同，零样本 CoT 没有在提示中加入人工标注的样例。相反，它直接生成推理步骤，然后利用生成的 CoT 来得出答案。零样本 CoT 最初是在 [250] 中提出的，其中大语言模型首先通过 "*Let's think step by step*" 提示生成推理步骤，然后通过 "*Therefore, the answer is*" 提示得出最终答案。他们发现，这种策略在模型规模超过一定大小时可以显著提高性能，但在小型模型中效果不佳，这是涌现能力的表现。为了在更多任务上解锁 CoT 能力，Flan-T5 和 Flan-PaLM [83] 使用 CoT 进行了指令调整，有效增强了在未见任务上的零样本性能。

6.2.2 关于 CoT 的讨论

在这部分中，我们将讨论两个与 CoT 相关的基本问题，即 "*CoT* 何时适用于大语言模型" 和 "大语言模型为什么能够进行 *CoT* 推理"。

CoT 何时适用于大语言模型? 由于 CoT 是一种涌现能力 [47]，它只能有效增强有 100 亿或更多参数的足够大的模型 [32]，而对小模型则无效。此外，由于 CoT 通过中间推理步骤增强了标准提示，它的效果主要体现在需要逐步推理的任务 [32]，例如算术推理、常识推理和符号推理。然而，对于不依赖于复杂推理的其他任务，它可能会比标准提示表现更差 [252]，例如 GLUE 数据集 [256] 中的 MNLI-m/mm、SST-2 和 QQP。有趣的是，CoT 提示带来的性能提升似乎只有在标准提示表现较差的情况下才会较为显著 [32]。

大语言模型为什么能够进行 CoT 推理? 我们将从以下两个方面讨论 CoT 的基本机制。

• *CoT* 能力的来源。关于 CoT 能力的来源，研究者普遍将其归因于使用代码进行训练，因为在代码数据训练过的模型表现出强大的推理能力 [46, 257]。直观上，代码数据具有规范的算法逻辑和流程，这可能有助于提高大语言模型的推理性能。然而，这个假设仍然缺乏公开报告的消融实验作为证据（有和没有代码训练）。此外，指令调整似乎不是获得 CoT 能力的关键原因，因为有实验表明，在非 CoT 数据上进行指令调整不会提高模型使用 CoT 完成任务的性能 [83]。

• 各个组件的作用。 CoT 提示与标准提示之间的主要区别在于在最终答案之前加入了推理路径，因此一些研究人员调查了推理路径中不同组成部分的影响。具体而言，最近的一项研究首先定义了 CoT 提示中的三个关键组成部分，即符号（例如算术推理中的数字）、模式（例如算术推理中的方程）和文本（即不是符号或模式的其余词）[258]。结果表明，后两部分（即模式和文本）对模型的性能至关重要，去除其中任何一部分都会导致性能显著下降。然而，符号和模式的正确性似乎并不关键。此外，文本和模式之间存在共生关系：文本有助于大语言模型生成有用的模式，而模式则可以帮助大语言模型理解任务并生成额外文本以帮助解决任务 [258]。

总之，CoT 提示提供了一种通用而灵活的方法来引出大语言模型的推理能力。还有一些工作尝试将这种技术扩展至多模态任务 [259] 和多语言任务 [260]。除了直接通过 ICL 和 CoT 使用大语言模型，最近还有一些研究探索了如何将大语言模型的能力用于特定任务 [261–263]，这被称为模型专业化 [264]。例如，在 [264] 中，研究人员用大语言模型生成推理

路径，然后再用这些推理路径微调小规模的语言模型 Flan-T5 [83]，从而将大语言模型的数学推理能力专业化。模型专业化可以应用于解决各种任务，如问答 [265]、代码生成 [266] 和信息检索 [267]。

7 能力评测

为了检验大语言模型（LLMs）的有效性和优越性，已有研究采用了大量的任务和基准数据集来进行实证评估和分析。首先，我们介绍了大语言模型在语言生成和语义理解方面的三种基本评估任务。然后，介绍了大语言模型在几种更复杂的设置或目标下的高级任务。最后，讨论了现有的基准和实证分析。

7.1 基础评测任务

在本部分中，我们主要关注大语言模型的三种评估任务，即语言生成、知识利用和复杂推理。需要注意的是，我们并不打算对所有相关任务进行完整覆盖，而是只关注大语言模型领域中最广泛讨论或研究的任务。接下来，我们将详细介绍这些任务。

7.1.1 语言生成

根据任务定义，现有语言生成的任务主要可以分为语言建模、条件文本生成和代码合成任务。需要注意的是，代码合成不是典型的自然语言处理任务，但这类任务可以通过一些大语言模型（经过代码数据训练的模型）以类似自然语言文本生成的方法来解决，因此我们也将其纳入讨论。

语言建模：作为大语言模型最基本的能力，*语言建模*旨在基于前面的词元预测下一个词元 [15]，主要关注基本的语言理解和生成能力。用于评估这种能力的典型语言建模数据集包括 Penn Treebank [268]、WikiText-103 [269] 和 Pile [117]，其中*困惑度*指标通常用于评估零样本情况下模型的性能。实证研究 [55, 82] 表明，LLMs 在这些评估数据集上相较于之前的最先进方法带来了实质性的性能提升。为了更好地测试文本中的长距离依赖建模能力，引入了 LAMBADA 数据集 [155]，其中要求大语言模型基于一段上下文来预测句子的最后一个单词。然后使用预测的最后一个单词的准确性和困惑度来评估大语言模型性能。正如现有工作所示，语言建模任务的性能通常遵循扩展定律 [30]，这意味着提升语言模型的参数量将提高准确性并降低困惑度。

条件文本生成：作为语言生成中的一个重要话题，条件文本生成 [48] 旨在基于给定的条件生成满足特定任务需求的文本，通常包括机器翻译 [337]、文本摘要 [338] 和问答系统 [339] 等。为了衡量生成文本的质量，通常使用自动指标（如准确率、BLEU [340] 和 ROUGE [341]）和人类评分来评估性能。由于大语言模型具有强大的语言生成能力，它们在现有的数据集上取得了显著的性能，甚至超过了人类表现（在测试数据集上）。例如，仅给出 32 个示例作为输入，GPT-3 通过上下文学习能够在 SuperGLUE 的平均得分上超过使用完整数据微调的 BERT-Large [282]；在 MMLU 上，一个 5-shot 的 Chinchilla [33] 的准确率几乎比人类的平均准确率提高了一倍，而在 5-shot 设置下，GPT-4 [45] 取得了当前最优秀的性能，平均准确率比之前的最佳模型提高了超过 10%。于是，人们开始关注现有的条件文本生成任务是否能够适当地评估和反映大语言模型的能力。考虑到这个问题，研究人员试图通过收集目前无法解决的任务（即大语言模型无法取得良好表现的任务）或创建更具挑战性的任务（例如超长文本生成 [342]）来制定新的评估基准，例如 BIG-bench Hard [284]。此外，最近的研究还发现自动指标可能会低估大语言模型的生成质量。在 OpenDialKG [281] 中，ChatGPT 在 BLEU 和 ROUGE-L 指标上表现不如微调的 GPT-2，但在人类评分中获得了更多的好评 [343]。因此，需要更多的努力来开发更符合人类偏好的新指标。

代码合成：除了生成高质量的自然语言外，现有的大语言模型还表现出强大的生成形式化语言的能力，尤其是满足特定条件的计算机程序（即代码），这种能力被称为*代码合成* [344]。与自然语言生成不同，由于生成的代码可以通过相应的编译器或解释器直接进行检查，现有的工作主要通过计算测试用例的通过率（即 pass@k）来评估大语言模型生成的代码的质量[16]。最近，有工作提出了几个专注于功能正确性的代码基准，用来评估大语言模型的代码合成能力，例如 APPS [286]、HumanEval [88] 和 MBPP [140]。通常，它们由各种编程问题组成，具有题目描述和用于检查正确性的测试用例。为了提高这种能力，在代码数据上微调（或预训练）大语言模型是关键所在，这可以有效地使 LLMs 适应代码合成任务 [76]。此外，现有的工作提出了新的代码生成策略，例如采样多个候选解 [140] 和规划引导的解码 [345]，这可以被认为是模仿程序员修复错误和规划代码编写的过程。令人印象深刻的是，大语言模型最近在程序竞赛平台 Codeforces 上取得了所有选手中前 28% 的排名，与人类表现相当 [98]。此外，GitHub Copilot 已发布，可在编程 IDE（如 Visual Studio 和 JetBrains IDE）中协助编程，支持包括 Python、JavaScript 和 Java 在内的多种语言。ACM 通讯中的一篇观点文章“*编程的终结*” [346] 讨论了 AI 编程在计算机科学领域的影响，强调了一个重要的转变，即将高度适应的大预言模型作为新的计算原子单位。

主要问题：虽然大语言模型在生成类似于人类的文本已经取得了出色的表现，但它们容易受到以下两个语言生成方面的问题影响。

- *可控生成*：对于大语言模型，生成给定条件下的文本的主流方法，是通过使用自然语言指令或提示进行。尽管这种机制很简单，但对这些模型生成输出的细粒度或结构限制

16. 给定大语言模型生成的 k 个程序，当至少有一个程序通过所有测试用例时，pass@k 被计算为 1，否则为 0

表 6
大语言模型的基础评测任务和相应的代表性数据集。

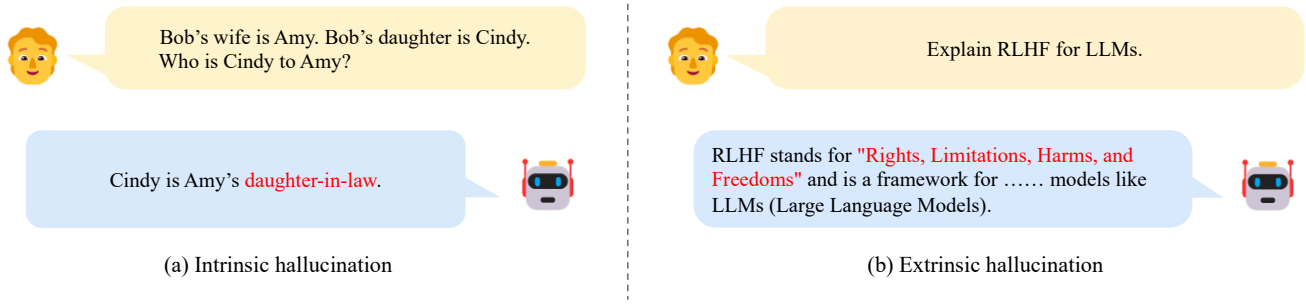
Task		Dataset
Language Generation	Language Modeling	Penn Treebank [268], WikiText-103 [269], the Pile [117], LAMBADA [155]
	Conditional Text Generation	WMT'14,16,19,20,21,22 [270–275], Flores-101 [276], DiaBLa [277], CNN/DailyMail [278], XSum [279], WikiLingua [280], OpenDialKG [281] SuperGLUE [282], MMLU [283], BIG-bench Hard [284], CLUE [285]
	Code Synthesis	APPS [286], HumanEval [88], MBPP [140], CodeContest [98], MTPB [76], DS-1000 [287], ODEX [288]
Knowledge Utilization	Closed-Book QA	Natural Questions [289], ARC [290], TruthfulQA [291], Web Questions [292], TriviaQA [293], PIQA [294], LC-quad2.0 [295], GrailQA [296], KQApro [297], CWQ [298], MKQA [299], ScienceQA [300]
	Open-Book QA	Natural Questions [289], OpenBookQA [301], ARC [290], Web Questions [292], TriviaQA [293], MS MARCO [302], QASC [303], SQuAD [304], WikiMovies [305]
	Knowledge Completion	WikiFact [306], FB15k-237 [307], Freebase [308], WN18RR [309], WordNet [310], LAMA [311], YAGO3-10 [312], YAGO [313]
Complex Reasoning	Knowledge Reasoning	CSQA [246], StrategyQA [247], ARC [290], BoolQ [314], PIQA [294], SIQA [315], HellaSwag [316], WinoGrande [317], OpenBookQA [301], COPA [318], ScienceQA [300], proScript [319], ProPara [320], ExplaGraphs [321], ProofWriter [322], EntailmentBank [323], ProOntoQA [324]
	Symbolic Reasoning	CoinFlip [32], ReverseList [32], LastLetter [32], Boolean Assignment [325], Parity [325], Colored Object [326], Penguins in a Table [326], Repeat Copy [327], Object Counting [327]
	Mathematical Reasoning	MATH [283], GSM8k [243], SVAMP [244], MultiArith [328], ASDiv [245], MathQA [329], AQUA-RAT [330], MAWPS [331], DROP [332], NaturalProofs [333], PISA [334], miniF2F [335], ProofNet [336]

方面仍面临着重大挑战。现有工作 [40] 表明，当生成文本的时候施加复杂的结构约束，大语言模型可以很好地处理*局部关系*（例如，相邻句子之间的交互），但可能难以应对*全局关系*（即长距离相关性）。例如，要生成一个由多个段落组成的复杂长篇文章，仍然很难直接在全局上确保特定的文本结构（例如概念的顺序和逻辑流程）。对于需要遵循结构化规则或语法的生成任务，例如代码合成，会更加具有挑战性。为了解决这个问题，一种潜在的解决方案是将一次性生成（即直接生成目标输出）扩展到大语言模型的迭代提示。这模拟了人类写作过程，将语言生成分解成多个步骤，例如规划、起草、重写和编辑 [342]。几项研究已经证明，迭代提示可以激发相关知识，从而在子任务中实现更好的性能 [347, 348]。本质上，CoT 提示利用了将复杂任务分解成多步推理链的思想。此外，生成的文本的安全控制对于实际部署来说非常非常重要。研究表明大语言模型可能会生成包含敏感信息或冒犯性表达的文本 [45]。虽然 RLHF 算法 [61] 可以在一定程度上缓解这个问题，但它仍然依赖于相当数量的人工标注数据来微调大语言模型，缺乏客观的优化目标。因此，必须探索有效的方法来克服这些限制，实现对大语言模型的输出进行更安全的控制。

● *专业化生成*：尽管大语言模型已经学习了一般的语言模式以生成连贯的文本，但在处理专业领域或任务时，它们的生成能力可能受到限制。例如，一个已经在一般网络文章上进行训练的语言模型，在生成一个涉及许多医学术语和方法的医学报告时可能会面临挑战。直观上，领域知识对于模型的专业化至关重要。然而，将这种专业知识注入到大语言模型中并不容易。正如最近的分析所讨论的 [46, 349]，当大语言模型被训练以展现某些特定的能力，使它们在某些领域表现出色时，它们可能会在其他领域遇到困难。这样的问题与神经网络训练中的*灾难性遗忘* [350, 351] 有关，它指的是整合新旧知识时产生的冲突现象。类似的情况也出现在人类对大语言模型的对齐中，其中必须支付 "*对齐成本*" [61]（例如在上下文学习能力方面潜在的损失）以对齐于人类的价值观和需求。因此，开发有效的模型专业化方法至关重要，这些方法可以灵活地使大语言模型适应各种任务场景，并尽可能保留其原有的能力。

7.1.2 知识利用

知识利用是智能系统基于事实证据的支撑，完成知识密集型任务的重要能力（例如常识问题回答和事实补全）。具体而言，它要求大语言模型在必要的时候，适当地利用来自预训练语料库的丰富事实知识或检索外部数据。特别地，问答和知识补全已经成为评估这一能力的两种常用任务。根据测试任务



(a) Intrinsic hallucination

(b) Extrinsic hallucination

图 7. 一个开放大语言模型的内在和外在幻觉的例子（访问日期：2023 年 3 月 19 日）。作为内在幻觉的例子，大语言模型对 Cindy 和 Amy 之间的关系给出了错误的判断，这与输入相矛盾。对于外在的幻觉，在这个例子中，大语言模型似乎对 RLHF（从人类反馈中强化学习）的含义有不正确的理解，尽管它能正确理解 LLM 的含义（在本例中）。

（问答或知识补全）和评估设置（有或无外部资源），我们将现有的知识利用任务分为三种类型，即闭卷问答，开卷问答[17]和知识补全。

闭卷问答：闭卷问答任务 [352] 测试大语言模型从预训练语料库中获取的事实知识。大语言模型只能基于给定的上下文回答问题，不能使用外部资源。为了评估这一能力，可以利用几个数据集，包括 Natural Questions [289]、Web Questions [292] 和 TriviaQA [293]，其中广泛采用准确性作为指标。实验表明，大语言模型在这种情况下表现良好，甚至与最先进的开放领域问答系统的表现相匹配 [56]。此外，大语言模型在闭卷问答任务上的性能也显示出扩展定律的模式，包括模型大小和数据大小方面的扩展定律：增加参数和训练词元数量可以增加大语言模型的容量，并帮助它们从预训练数据中学习（或记忆）更多的知识 [56]。此外，在相似的参数规模下，通过更多与评估任务相关的数据训练的大语言模型将实现更好的性能 [71]。此外，闭卷问答设置还为探究大语言模型编码的事实知识的准确性提供了测试平台。然而，正如现有研究所示 [55]，即使在预训练数据中存在相关的知识，大语言模型在依赖于细粒度知识的问答任务上的表现可能也会较差。

开卷问答：与闭卷问答不同，开卷问答任务中，大语言模型可以从外部知识库或文档集合中提取有用的证据，然后基于提取的证据回答问题 [353–356]。典型的开卷问答数据集（例如，Natural Questions [289]、OpenBookQA [301] 和 SQuAD [304]）。虽然开卷问答数据集与闭卷问答数据集有重叠，但它们包含外部数据源，例如维基百科。准确性和 F1-score 是开卷问答任务中广泛使用的评估指标。为了从外部资源中选择相关知识，大语言模型通常与一个文本检索器（甚至是一个搜索引擎）配对，该文本检索器与大语言模型独立或联合进行训练 [71, 353, 357]。在评测的过程中，现有研究主要关注并测试大语言模型如何利用提取的知识回答问题，并表明检索到的证据可以大大提高生成答案的准确性，甚至使较小的语言模型能够胜过比其参数量大 10 倍的大语言模型 [353, 357]。此外，开卷问答任务还可以评估知识信息的新旧程度。从过时的知识资源进行预训练或检索，可能会导致大语言模型对于时间敏感的问题生成不正确的答案 [353]。

知识补全：在知识补全任务中，大语言模型在某种程度上可以被视为一个知识库 [311]，可以用于补全或预测知识单元（例如，知识三元组）的缺失部分。这种任务可以探究和评估大语言模型从预训练数据中学习的多少和什么样的知识。现有的知识补全任务可以粗略地分为知识图谱补全任务（例如，FB15k-237 [307] 和 WN18RR [309]）和事实补全任务（例如，WikiFact [306]），分别旨在补全知识图谱中的三元组和补全有关特定事实的句子。经验证实，现有的大语言模型很难完成涉及特定关系类型的知识完成任务 [257]。在 WikiFact 的评估结果中，大语言模型在预训练数据中出现频率较高的一些关系（例如 `currency` 和 `author`）上表现良好，但在出现较少的关系（例如 `discoverer_or_inventor` 和 `place_of_birth`）上表现不佳。有趣的是，在相同的评估设置下（例如上下文学习），InstructGPT（即 `text-davinci-002`）在 WikiFact 的所有子集中均优于 GPT-3。这表明，指令微调有助于大语言模型完成知识完成任务。

主要问题：尽管大语言模型在捕获和利用知识信息方面取得了关键进展，但它们存在以下两个主要问题。

- *幻觉（Hallucination）*：在生成事实文本时，一个具有挑战性的问题是*幻觉生成* [343]，即其生成的信息与现有来源相冲突（*内在幻觉*）或无法通过现有来源验证（*外在幻觉*）。图 7 中展示了两个例子。幻觉在现有的大语言模型中广泛存在，甚至包括最优秀的大语言模型，如 GPT-4 [45]。实质上，大语言模型似乎“无意识地”在解决任务的过程中利用中的知识，但仍缺乏准确控制内部或外部知识使用的能力。幻觉会误导大语言模型生成不良输出，大大降低性能，在实际应用中可能带来潜在风险。为了缓解这个问题，现有的工作广泛利用对齐调整策略（如第 5.2 节中讨论的），依赖于在高质量数据上

17. 在本部分中，开卷问答是指需要从外部知识资源中提取和利用有用信息的问答任务，是闭卷问答（仅使用预训练语料库中的编码信息）的对立面。请注意，还有一个名为 OpenBookQA 的数据集 [301]，它遵循开卷问答任务的设置，通过提取和利用外部科学事实来回答问题。

或使用人类反馈对大语言模型进行微调。为了评估幻觉问题，已经提出了一系列幻觉检测任务，例如 TruthfulQA [291]，用于检测模型是否会模仿人类的虚假言论。

● 知识新颖性：作为另一个主要挑战，大语言模型在解决需要使用比训练数据更新的知识的任务时会遇到困难。为了解决这个问题，一个直接的方法是定期用新数据更新大语言模型。然而，微调大语言模型是非常昂贵的，而且在增量训练大语言模型时很可能会导致灾难性遗忘问题。因此，有必要开发高效有效的方法，将新知识集成到现有的大语言模型中，使其保持最新状态。现有的研究已经探索了如何利用外部知识源（例如搜索引擎）来补充大语言模型，这可以是与大语言模型一起优化的 [353]，也可以是作为一种即插即用的模块 [358]。例如，ChatGPT 利用检索插件访问最新的信息源 [359]。通过将提取的相关信息并入上下文 [360, 361]，大语言模型可以获取新的事实知识，并在相关任务上有更好的表现。然而，这种方法似乎仍然处于表面层次。实验已经揭示了，直接修改内在知识或将特定知识注入大语言模型是很困难的，这仍然是一个值得研究的研究问题 [362, 363]。

7.1.3 复杂推理

复杂推理是指理解和利用相关的证据或逻辑来推导结论或做出决策的能力 [51, 52]。根据推理过程中涉及的逻辑和证据类型，我们考虑将现有的评估任务分为三个主要类别，即知识推理、符号推理和数学推理。

知识推理： 知识推理任务依赖于事实知识的逻辑关系和证据来回答给定的问题。现有的工作主要使用特定的数据集来评估相应类型的知识的推理能力，例如 CSQA [246]/StrategyQA [247] 用于常识推理，ScienceQA [300] 用于科学知识推理。此外，现有的工作 [300] 还通过自动化评测（例如 BLEU）或人类评估来评估生成的推理过程的质量。通常，这些任务要求大语言模型根据事实知识逐步推理，直到回答给定的问题。为了激发逐步推理的能力，有研究提出了思维链 (Chain-of-Thoughts, CoT) 提示策略 [32] 来增强大语言模型的复杂推理能力。如第 6.2 节所述，CoT 包括中间推理步骤，可以手动创建 [32] 或自动生成 [364]，以指导大语言模型进行多步推理。这种方式大大提高了大语言模型的推理性能，使得在几个复杂知识推理任务上取得了最佳的效果 [32, 56, 365]。此外，将知识推理任务转化为代码生成任务后，研究人员发现大语言模型的性能可以进一步提高 [144]，特别是对于在代码上预训练的大语言模型。然而，由于知识推理任务的复杂性，当前大语言模型的性能仍然落后于人类在常识推理等任务上取得的结果。作为最常见的错误之一，大语言模型可能基于错误的事实知识生成不准确的中间步骤，导致错误的最终结果。为了解决这个问题，现有的工作提出了特殊的解码或投票策略来提高整个推理链的准确性。最近的一项实证研究 [365] 表明，大语言模型可能难以明确推断出特定任务所需的常识知识，尽管它们可以成功地解决该任务。此外，它进一步表明，利用自动生成的知识可能不利于提高推理性能。

符号推理[18]**：** 符号推理任务主要关注于在形式化规则设置中操作符号以实现某些特定目标 [51]，其中操作和规则可能在大语言模型预训练期间从未被看到过。现有的工作 [32, 232, 250] 通常在尾字母拼接和硬币反转任务上评估大语言模型，其中用于评测的数据与上下文示例有相同的推理步骤（称为领域内测试）或更多步骤（称为领域外测试）。对于领域外测试的例子，大语言模型只能看到上下文示例中有两个单词的示例，但需要大语言模型在测试中需要将三个或更多单词的最后一个字母进行拼接。通常，采用所生成符号的准确性来评估大语言模型在这些任务上的性能。因此，大语言模型需要理解符号操作之间的语义关系以及它们在复杂场景中的组合。然而，在领域外测试下，由于大语言模型没有看到符号操作和规则的复杂组合（例如上下文示例中有两倍的操作数量），因此难以捕捉其准确含义。为了解决这个问题，现有研究结合了 scratchpad [325, 366] 和 tutor [367] 策略来帮助大语言模型更好地操作符号，生成更长和更复杂的推理过程。另一条研究路线利用形式化编程语言来表示符号操作和规则，这要求大语言模型生成代码并通过外部解释器执行推理过程。这种方法可以将复杂的推理过程分解为大语言模型的代码合成和解释器的程序执行，从而简化推理过程并获得更准确的结果 [327]。

数学推理： 数学推理任务需要综合利用数学知识、逻辑和计算来解决问题或生成证明过程。现有的数学推理任务主要可分为数学问题求解和自动定理证明两类。对于数学问题求解任务，常用的评估数据集包括 SVAMP [244]、GSM8k [243] 和 MATH [283] 数据集，其中大语言模型需要生成准确的具体数字或方程来回答数学问题。由于这些任务也需要多步推理，CoT 提示策略已被广泛采用来提高大语言模型的推理性能 [32]。作为一种实际策略，持续在大规模数学语料库上预训练大语言模型可以大大提高它们在数学推理任务上的性能 [34, 135, 368]。此外，由于不同语言中的数学问题共享相同的数学逻辑，研究人员还提出了一个多语言数学问题基准测试 [260]，用于评估大语言模型的多语言数学推理能力。作为另一个具有挑战性的任务，自动定理证明（ATP）[333, 335, 369] 需要推理模型严格遵循推理逻辑和数学技能。为了评估在此任务上的性能，PISA [334] 和 miniF2F [335] 是两个典型的 ATP 数据集，其中证明成功率是评估指标。作为一种典型的方法，现有的 ATP 工作利用大语言模型来辅助定理证明器（如 Lean、Metamath 和 Isabelle）进行证明搜索 [370–372]。ATP 研究的一个主要限制是缺乏形式语言相关的语料库。为了解决这个问题，一些研究利用大语言模型将非形式化语句

18. 我们主要讨论特别设计用于评估大语言模型的符号推理任务，不考虑传统自然语言处理任务中的符号推理方法，例如从知识图谱中推导逻辑规则的 KBQA。

转换为形式证明以增加新数据 [145]，或者生成草稿和证明草图以减少证明搜索空间 [373]。

主要问题：尽管有所进展，大语言模型在解决复杂的推理任务方面仍存在一些限制。

- 不一致性：通过改进推理策略（如 CoT），大语言模型可以执行基于逻辑和支撑性证据的逐步推理，从而解决一些复杂的推理任务。尽管这种方法是有效的，但在推理过程中经常出现 不一致性问题。具体而言，大语言模型可能会在错误的推理路径下生成正确答案，或者在正确的推理过程之后产生错误答案 [32, 374]，导致导出的答案与推理过程之间存在不一致性。为了缓解这个问题，现有的工作提出了通过外部工具或模型指导大语言模型的整个生成过程 [345]，或者重新检查推理过程和最终答案以进行纠正 [375] 的方法。作为一种潜在的解决方案，最近的方法将复杂的推理任务重新转化为代码生成任务，其中通过严格执行生成的代码以确保了推理过程和结果之间的一致性。此外，还发现可能存在相似输入的任务之间的不一致性，其中任务描述中的微小变化可能会导致模型产生不同的结果 [49, 244]。为了缓解这个问题，可以使用多个推理路径的进行投票来增强大语言模型的解码过程 [251]。
- 数值计算：对于复杂的推理任务，大语言模型在涉及数值计算时仍然面临困难，特别是对于在预训练阶段很少遇到的符号，例如大数字的算术运算 [49, 367]。为了解决这个问题，一种直接的方法是在合成的算术问题上微调大语言模型 [376]。一系列的研究根据这种方法，并通过特殊的训练和推理策略进一步提高数值计算性能 [366]，例如使用草稿纸推演。此外，现有的工作还包括使用外部工具（例如计算器）来处理算术运算 [70]。最近，ChatGPT 提供了一个插件机制来使用外部工具 [359]。这样，大语言模型需要学习如何正确地操作这些工具。为此，研究人员通过使用工具（甚至是大语言模型本身）来调整大语言模型的示例 [70, 377]，或者为上下文学习设计指令和示例 [327]。然而，这些大语言模型仍然依赖于文本上下文来捕捉数学符号的语义含义（在预训练阶段），这本质上并不适合于数值计算。

7.2 高级能力评估

除了上述基本评估任务外，大语言模型还展现出一些需要特殊考虑的高级能力。在本节中，我们将讨论几种代表性的高级能力及其相应的评估方法，包括人类对齐、与外部环境的互动、工具操作等。接下来，我们将详细讨论这些高级能力。

7.2.1 人类对齐

人类对齐指的是让大语言模型能够很好地符合人类的价值和需求，这是在现实世界应用中广泛使用大语言模型的关键能力。

为了评估这种能力，现有的研究考虑了多个人类对齐的标准，例如有益性、诚实性和安全性 [45, 207, 208]。对于有益性和诚实性，可以利用对抗性问答任务（例如 TruthfulQA [291]）来检查大语言模型在检测文本中可能的虚假性方面的能力 [45, 71]。此外，有害性也可以通过若干现有的基准测试来评估，例如 CrowS-Pairs [378] 和 Winogender [379]。尽管存在以上数据集的自动评估，人工评估仍然是一种更直接有效的测试大语言模型人类对齐能力的方法。OpenAI 邀请了许多与 AI 风险相关的领域专家来评估和改进 GPT-4 在遇到风险内容时的行为 [45]。此外，对于人类对齐的其他方面（例如真实性），一些研究提出使用具体指令和设计标注规则来指导评价过程 [71]。实证研究表明，这些策略可以大大提高大语言模型的人类对齐能力 [208]。例如，在与专家交互收集的数据的对齐调整后，GPT-4 在处理敏感或不允许的提示时的错误行为率可以大大降低。此外，高质量的预训练数据可以减少对齐所需的工作量 [45]。例如，Galactica 模型在含有较少偏见内容的科学语料库上进行预训练，因此可能更加无害 [34]。

7.2.2 与外部环境的互动

除了标准评估任务外，大语言模型还具有从外部环境接收反馈并根据行为指令执行操作的能力，例如生成自然语言行动计划以操作智能体 [380, 381]。大语言模型中具备这种能力，可以生成详细且高度逼真的行动计划，而较小的模型（例如 GPT-2）倾向于生成较短或无意义的计划 [380]。

为了测试这种能力，研究者提出了一些具身体感知的人工智能基准进行评估。VirtualHome [382] 构建了一个 3D 模拟器，用于家务任务（例如清洁和烹饪），代理人可以执行大语言模型生成的自然语言行动。ALFRED [383] 包括更具挑战性的任务，需要大语言模型完成组合目标。BEHAVIOR [384] 侧重于在模拟环境中进行日常杂务，并要求大语言模型生成复杂的解决方案，例如更改对象的内部状态。对于大语言模型生成的行动计划，现有的工作要么采用基准测试中的常规指标（例如生成的行动计划的可执行性和正确性），要么直接根据现实世界执行的成功率来评估这种能力 [380, 385]。现有的工作已经显示出大语言模型在与外部环境的互动和生成准确的行动计划方面的有效性 [386]。最近，一些工作提出了几种改进方法来增强大语言模型的交互能力，例如设计类似代码的提示 [387] 和提供真实世界的反馈 [385]。

7.2.3 工具操作

在解决复杂问题时，大语言模型可以在必要的情况下利用外部工具。通过封装可用工具的 API 调用，现有的工作已经考虑了各种外部工具，例如搜索引擎 [71]、计算器 [70] 和编译器 [327] 等等，以增强大语言模型在特定任务上的性能。最近，OpenAI 已经支持在 ChatGPT 中使用插件 [359]，这使得大语言模型除了语言建模之外还具备了更广泛的能力。例如，Web 浏览器插件使 ChatGPT 能够访问实时信息。此外，整合第三方插件对于创建基于大语言模型的应用程序生态系统非常关键。

为了检验工具操作的能力，现有的工作大多采用复杂的推理任务进行评估，例如数学问题求解（例如 GSM8k [243] 和 SVAMP [244]）或知识问答（例如 TruthfulQA [291]），其中成功操作工具对于增强大语言模型缺乏的所需技能非常重要（例如数值计算）。通过这种方式，这些任务的评估性能可以反映出大语言模型在工具操作方面的能力。为了让大语言模型学会利用工具，现有研究在上下文中添加使用工具的示例来让大语言模型学习使用方法 [327]，或基于工具操作的相关数据对大语言模型进行微调 [70, 377]。现有的工作发现，在工具的帮助下，大语言模型变得更加能够解决它们不擅长的问题，例如方程计算和利用实时信息，并最终提高了最终的性能 [70]。

总结. 上述三种能力对于大语言模型在实际应用中的表现具有巨大的价值：符合人类价值和偏好（人类对齐）、在实际场景中正确行动（与外部环境交互）和扩展能力范围（工具操作）。除了上述三种高级能力之外，大语言模型还可能展现出一些有关特定任务（例如数据标注 [227]）或学习机制（例如自我改进 [255]）的其他高级能力。发现、衡量和评估这些新兴能力以更好地利用和改进大语言模型将是一个开放的研究方向。

7.3 Public Benchmarks and Empirical Analysis

在上述章节中，我们已经讨论了大语言模型的评估任务及其相应的设置。接下来，我们将介绍现有的大语言模型评测基准和实证分析，从总体视角对大模型的能力进行更全面的讨论。

7.3.1 评测基准

数个用于评估大语言模型的综合性评测基准已于近日发布 [257, 283, 326] 。在本节中，我们将介绍几个具有代表性并得到广泛使用的评测基准，即 MMLU、BIG-bench 和 HELM。

• *MMLU* [283] 是一个通用评测基准，用于大规模评测大语言模型的多任务知识理解能力。其涉及到的知识涵盖数学，计算机科学以及人文和社会科学等领域，并包含从基础到进阶不同难度的任务。现有工作表明，大语言模型在这个基准上大多数情况下比小模型表现出更高的性能 [34, 56, 57, 83]，这表明了模型尺寸的规模定律。最近，GPT-4 在 MMLU 上取得了显著成果（5-shot 设置下正确率达到 86.4%），远远优于以前的最佳模型 [45]。

• *BIG-bench* [326] 是一个由社区协作收集的评测基准，旨在从各个方面探究现有大语言模型的能力。它包含了 204 个任务，主题包括语言学、儿童发展、数学、常识推理、生物学、物理学、社会偏见、软件开发等等。通过扩展模型尺寸，few-shot 设置下的大语言模型甚至可以在 65% 的 BIG-bench 任务中超过平均人类表现 [56]。鉴于该评测基准的高评估成本，作者还提出了一个轻量级基准 BIG-bench-Lite，其中包含来自 BIG-bench 的 24 个小型、多样且具有挑战性的任务。此外，研究者们从 BIG-bench 中挑选大语言模型表现劣于人类的挑战性任务，提出了 BIG-bench hard（BBH）基准，用以探索大语言模型当前无法解决的任务。实验发现随着任务难度的增加，大部分小模型的性能接近于随机猜测。相比之下，思维链提示可以引出大语言模型逐步推理的能力从而增强性能，使其在 BBH 中超过平均人类表现 [284]。

• *HELM* [257] 是一个综合性评测基准，目前包括 16 个核心场景和 7 类指标。它建立在许多先前提出的评测基准之上，旨在对大语言模型进行全面评估。HELM 的实验结果显示，指令微调可以在准确性、鲁棒性和公平性方面提高大语言模型的性能。此外，对于推理任务，已经在代码语料库上预训练的大语言模型表现出更优秀的性能。

以上评测基准涵盖了大量的主流大语言模型评估任务。此外，还有一些评测基准专注于评估大语言模型在特定任务上的能力，如用于评测多语言知识利用能力的 TyDiQA [388] 和用于评测多语言数学推理的 MGSM [260]。研究者可以根据想要评测的能力选择相应基准。此外，Language Model Evaluation Harness [389] 和 OpenAI Evals [45] 等开源评估框架可供研究人员在现有评测基准上评估大语言模型，或者在新任务上进行个性化评估。

7.3.2 大语言模型能力的综合分析

除了构建大规模评估基准之外，大量研究已经进行了全面分析，以调查大语言模型的优势和局限性。在本部分中，我们将主要从两个主要方面简要讨论它们，即*通才*（通用能力）和*专才*（特定领域能力）。

通才： 由于大预言模型出色的表现，现有的工作 [40, 45, 343, 349, 390–392] 对它们的通用能力进行了系统的评估，以探索它们在众多任务或应用中的表现。这些研究主要关注之前未经过充分调查的新兴大语言模型（例如 ChatGPT 和 GPT-4），具体内容如下所述：

• *熟练度*：为了评估大语言模型在解决一般任务方面的熟练度，现有的工作 [392] 通常收集一组涵盖各种任务和领域的数据集，然后在 few-shot 或 zero-shot 设置下测试大语言模型的性能。实验结果 [40, 45, 349, 392] 显示大语言模型对通用任务有着卓越的解决能力。作为一项显著进展，GPT-4 在语言理解、常识推理和数学推理等一系列任务中超越了此前在特定数据集上训练过的方法 [45]。此外，它可以在为人类设计的真实世界考试（例如美国大学预修课程考试和研究生入学考试）中达到近似于人类的表现 [45]。最近，一项全面的定性分析 [40] 揭示了 GPT-4 能在各个领域的各种具有挑战性的任务中接近人类水平，例如数学、计算机视觉和编程，并将其视为“一个人工通用智能系统的早期版本”。在这些令人鼓舞的结果之外，该分析也表明 GPT-4 仍然具有严重的局限性。例如，GPT-4 难以校准生成结果的置信度，并且无法验证其与训练数据和自身的一致性。此外，几项研究还表明，大语言模型可能会误解陌生概念 [392, 393]，并且在解决与情感相关的

实用任务方面 [391]（例如个性化情感识别）面临挑战，表现不及特定的微调模型。

• 稳定度：对大模型的综合分析需要考虑的另一个方面是它们对噪声或扰动的稳定性，这对于实际应用尤其重要。为了评估大语言模型对噪声或扰动的稳定度，现有的工作 [394] 对输入进行对抗攻击处理（例如符号替换），然后根据输出结果的变化评估大语言模型的稳定性。实验表明大语言模型在各种任务中比小型语言模型更稳定，但也会遇到一些新的相关问题，例如稳定度的不一致性和提示词敏感。具体来说，对于具有相同含义而表达方式不同的输入，大语言模型往往会提供不同的答案，甚至与自身生成的内容相矛盾 [395]。这样的问题也会导致在使用不同提示词评估稳定性时产生不一致的结果，使稳定性分析的评估结果本身不太可靠。

专才：由于大语言模型已经在大规模语料库上进行了预训练，它们可以从预训练数据中获取丰富的知识。因此，大语言模型可以被用作特定领域的专家。最近的研究广泛探索了将大语言模型用于解决特定领域任务的应用，并评估了大语言模型的适应能力。通常，这些研究收集或构建特定领域的数据集，使用上下文学习来评估大语言模型的性能。由于我们的重点不是覆盖所有可能的应用领域，因此我们简要讨论了三个受到研究界广泛关注的代表性领域，即医疗、教育和法律。

• 医疗是一个与人类生命密切相关的重要应用领域。自 ChatGPT 问世以来，一系列研究已经将 ChatGPT 或其他大语言模型应用于医疗领域。大语言模型能够处理各种医疗保健任务，例如生物信息提取 [396]、医疗咨询 [397–399] 和报告简化 [400]，甚至可以通过为专业医生设计的医疗执照考试 [401–403]。然而，大语言模型可能会制造医学错误信息 [398, 400]，例如错误解释医学术语并提供与医学指南不一致的建议。此外，上传患者健康信息也会引起隐私问题 [396]。

• 教育也是一个重要的应用领域。已有研究发现，大语言模型可以在数学、物理、计算机科学等科目的标准化测试中达到学生级别的表现 [45, 404, 405]，这些测试包括选择题和开放式问题。此外，实验表明大语言模型可以作为写作或阅读助手 [406, 407]。最近的一项研究 [407] 表明，ChatGPT 可以生成在不同学科之间逻辑一致并且平衡深度和广度的答案。另一项定量分析 [406] 表明，在某些计算机安全领域的课程中，利用 ChatGPT 的学生表现比使用其他方法的学生的平均表现更好（例如保留或完善大语言模型结果作为自己的答案）。然而，大语言模型的普及也引发了关于如何合理使用这样的智能助手的担忧（例如如何避免作弊行为）。

• 法律是一个建立在专业知识之上的专业领域。最近的一些研究已经应用大语言模型来解决各种法律任务，例如法律文件分析 [408, 409]、法律判决预测 [410] 和法律文件撰写 [411] 。最近的一项研究 [412] 发现，大语言模型具有强大的法律解释和推理能力。此外，最新的 GPT-4 模型在模拟律师考试中取得了相当于人类考生前 10% 的成绩。然而，大语言模型在法律领域的使用也引发了关于法律挑战的担忧，包括版权问题 [413]、个人信息泄露 [414] 以及偏见和歧视 [415]。

在上述工作外，一些工作还从其他角度分析了大语言模型的能力。例如，一些工作研究了 LLMs 的类人特征，如自我意识、心理理论（Theory of Mind, ToM）和情感计算等方面的特征 [40, 416–418]。特别地，针对两个经典的虚假信念任务进行的 ToM 的实验表明，GPT-3.5 系列模型在 ToM 任务中的表现与 9 岁儿童相当，因此推测大语言模型可能具有类似 ToM 能力 [417]。此外，另一些工作调查了目前大语言模型的评估设置的公平性和准确性 [419]，例如，大规模的预训练数据可能包含测试集中的数据。

8 总结与未来方向

在这篇综述中，我们回顾了大语言模型（LLMs）的最新进展，并介绍了理解和利用 LLMs 的关键概念、发现和技术。我们重点关注大模型（即大小超过 10B 的模型），并未考虑与早期预训练语言模型（例如 BERT 和 GPT-2）的相关内容，因为它们已经在现有文献中得到了很好的综述。具体来说，我们的综述讨论了 LLM 的四个重要方面，即预训练、适配微调、应用和评估。针对每个方面，我们重点介绍了对 LLM 成功至关重要的技术或发现。此外，我们还总结了开发 LLM 的可用资源，并讨论了实现 LLM 的重要技术，以便复现 LLM。这篇综述试图涵盖关于 LLM 的最新文献，并为研究人员和工程师提供一份有关这个主题的优质参考资料。接下来，我们总结了本文的讨论，并在以下方面介绍了 LLM 的挑战和未来方向。

理论与原理：对于大语言模型（LLM）的基本工作机制，最大的谜题之一是其如何通过非常大且深的神经网络分配、组织和利用信息。揭示建立 LLM 能力基础的基本原则或要素非常重要。具体来说，扩展规模似乎在提高 LLM 的能力方面起着重要作用 [47, 55, 59]。已有工作显示，当语言模型的参数增加到某个临界规模（例如 100 亿）时，会以一种意想不到的方式（突然性能飞跃）涌现出一些能力 [32, 47]，通常包括上下文学习、指令跟随和逐步推理。这些涌现能力既令人着迷又令人困惑：何时和如何它们被 LLM 获得尚不清楚。最近的研究要么进行广泛的实验来调查涌现能力的影响和这些能力的贡献因素 [219, 236, 420]，要么用现有的理论框架解释一些具体能力 [60, 230]。一个富有洞察力的技术博客也以 GPT 系列模型为目标，专门讨论了这个话题 [46]。然而，更多能理解、描述和解释 LLM 的能力或行为的正式理论和原理仍然缺失。由于涌现能力与自然界的相变具有十分相似的类比关系 [47, 58]，跨学科理论或原则（例如，LLM 是否可以被视为某种复杂系统）可能对解释和理解 LLM 的行为有用。这些基本问题值得研究界探讨，且对于开发下一代 LLM 至关重要。

模型架构：堆叠的多头自注意力层组成的 Transformer，由于其可扩展性和有效性，已成为构建大语言模型（LLM）的基

本架构。已有方法已经提出了各种策略来提高该架构的性能，如神经网络配置和可扩展的并行训练（请参阅第 4.2.2节的讨论）。为了提高模型容量（例如多轮对话能力），现有的 LLM 通常维持一个较长的上下文窗口，例如 GPT-4-32k 的上下文长度达到了 32,768 个词。因此，减少标准自注意力机制所带来的时间复杂度（原始为二次代价）是一个实际应用时重要的考虑因素。研究如何构建 LLM 中更高效的 Transformer 变体十分重要 [421]，例如 GPT-3 中已经使用了稀疏注意力 [55]。此外，灾难性遗忘一直是神经网络的长期挑战，其对 LLM 也有负面影响。在使用新数据调整 LLM 时，原先学到的知识可能会受到损害，例如根据某些特定任务对 LLM 进行微调将影响 LLM 的通用能力。当 LLM 与人类价值观保持一致时（称为对齐税 [61, 207]），也会出现类似情况。因此，有必要考虑将现有架构扩展到更具灵活性的机制或模块，以有效支持数据更新和任务专用化。

模型训练： 在实践中，由于巨大的计算消耗和对数据质量和训练技巧的敏感性 [68, 82]，预训练功能强大的大语言模型非常困难。因此，开发更系统、经济的预训练方法以优化 LLM 变得尤为重要，同时考虑到模型有效性、效率优化和训练稳定性等因素。我们应该开发更多的模型检查或性能诊断方法（例如 GPT-4 中的可预测扩展 [45]），以便在训练过程中及早发现异常问题。此外，还需要更灵活的硬件支持或资源调度机制，以便更好地组织和利用计算集群中的资源。由于从头开始预训练 LLM 的成本非常高，因此设计适合的机制在公开可用的模型检查点基础上不断预训练或微调 LLM 是非常重要的（例如 LLaMA [57] 和 Flan-T5 [83]）。为此，需要解决许多技术问题，例如灾难性遗忘和任务专门化。然而，迄今为止，仍缺乏具有完整预处理和训练日志的 LLM 开源模型检查点（例如准备预训练数据的脚本）以进行复现。我们相信，在开源模型中报告更多技术细节对于 LLM 研究将具有很大价值。此外，开发更多有效引导模型能力的改进调优策略也很重要。

模型应用： 由于在实际应用中微调的成本非常高，提示已成为使用大型语言模型的主要方法。通过将任务描述和示例合并到提示中，上下文学习（一种特殊形式的提示）赋予了 LLM 在新任务上表现良好的能力，甚至在某些情况下胜过全数据微调模型。此外，为了提高复杂推理能力，已有工作提出了先进的提示技术，例如链式思维（CoT）策略，它将中间推理步骤包含在提示中。然而，现有的提示方法仍然存在以下几个不足之处。首先，提示设计时需要大量人力。自动生成有效提示以解决各种任务将非常有用。其次，一些复杂任务（例如形式证明和数值计算）需要特定的知识或逻辑规则，这些规则可能无法用自然语言很好地表达或通过示例演示。因此，开发更具信息量和灵活性的任务格式化方法以进行提示非常重要[19]。第三，现有的提示策略主要关注单轮性能。开发交互式提示机制（例如通过自然语言对话）来解决复杂任务是有用的，这已经被 ChatGPT 证明非常有用。

安全与对齐： 尽管具有强大的能力，大型语言模型（LLM）与小型语言模型在安全方面面临类似的挑战。例如 LLM 倾向于产生幻觉 [343]，这些文本看似合理，但可能在事实上是错误的。更糟糕的是，LLM 可能被有意的指令激发以产生有害的、有偏见的或有毒的文本以用于恶意系统，从而导致潜在的滥用风险 [55, 61]。为了详细讨论 LLM 的安全问题（例如隐私、过度依赖、虚假信息和影响操作），读者可以参考 GPT-3/4 技术报告 [45, 55]。作为避免这些问题的主要方法，可通过将人类纳入训练循环来开发良好对齐的 LLM，并使用基于人类反馈的强化学习（RLHF）[61, 100]。为了提高模型安全性，在 RLHF 过程中包含安全相关的提示也非常重要，正如 GPT-4 所示 [45]。然而，RLHF 严重依赖专业标注者的高质量人类反馈数据，这使得它在实践中难以适当实施。因此，有必要改进 RLHF 框架以减少人类标注者的工作量，并寻求更高效的、具有保证数据质量的标注方法，例如 LLM 可以用于辅助标注工作。最近，红队方法 [209, 210] 已经被采用来提高 LLM 的模型安全性，该方法利用收集到的对抗性提示来优化 LLM（即避免红队攻击）。此外，通过聊天获取人类反馈并直接将其用于自我改进的适当学习机制也具有重要意义。

应用与生态： 随着 LLMs 在解决各种任务方面表现出强大的能力，它们可以应用于广泛的现实世界应用（即遵循特定任务的自然语言指令）。作为一个显著的进步，ChatGPT 可能已经改变了人类获取信息的方式，这已在 “New Bing” 的发布中得到实现。在不远的将来，可以预见到 LLM 将对信息检索技术产生重大影响，包括搜索引擎和推荐系统。此外，智能信息助手的开发和使用将随着 LLM 的技术升级得到高度推广。从更广泛的范围来看，这波技术创新浪潮将产生一个以 LLM 为支持的应用生态系统（例如 ChatGPT 对插件的支持），这与人类生活息息相关。最后，LLM 的兴起为人工通用智能（AGI）的探索提供了启示。有望开发出比以往更智能的系统（可能具有多模态信号）。然而，在这一发展过程中，AI 安全应成为主要关注之一，即 AI 对人类产生好处而非坏处 [39]。

尾声: 本综述是由我们研究团队在一次讨论会上计划的，我们旨在总结大语言模型的最新进展，为我们的团队成员提供一份高度可读性的报告。第一稿于 2023 年 3 月 13 日完成，我们的团队成员尽最大努力以相对客观、全面的方式囊括有关 LLM 的相关研究。接着，我们进行了多次细致的写作和内容修订。尽管我们付出了巨大的努力，但这份综述仍远非完美：我们可能会遗漏重要的参考文献或主题，也可能存在不严谨的表述或讨论。由于空间有限，我们只能按照特定的选择标准在图 2和表 1中展示部分现有的 LLM。然而，我们在 GitHub 页面（https://github.com/RUCAIBox/LLMSurvey）上设置了更为宽松的模型选择标准，该页面将定期维护。我们将不断更新这份调查，并尽力提高质量。对于我们来说，综

19. 然而，解决这个问题的另一种方法似乎是在任务难以通过文本生成解决时调用外部工具，例如 ChatGPT 的插件。

述写作也是我们自己对 LLM 的学习过程。对于那些有建设性意见来改进这份调查的读者，欢迎在我们综述的 GitHub 页面上留言或直接给我们的作者发电子邮件。我们将根据收到的评论或建议进行修订，并在我们的综述中致谢为此做出建设性贡献的读者。

致谢

作者们感谢 Yankai Lin 和 Yutao Zhu 对本文的校对。自本文首次发布以来，我们收到了许多来自读者的宝贵意见。我们真诚地感谢给我们邮件并提出建设性建议和评论的读者：Tyler Suard, Damai Dai, Liang Ding, Stella Biderman, Kevin Gray, and Jay Alammar.

参考文献

[1] S. Pinker, *The Language Instinct: How the Mind Creates Language*. Brilliance Audio; Unabridged edition, 2014.

[2] M. D. Hauser, N. Chomsky, and W. T. Fitch, "The faculty of language: what is it, who has it, and how did it evolve?" *science*, vol. 298, no. 5598, pp. 1569–1579, 2002.

[3] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. LIX, no. 236, pp. 433–460, 1950.

[4] F. Jelinek, *Statistical Methods for Speech Recognition*. MIT Press, 1998.

[5] J. Gao and C. Lin, "Introduction to the special issue on statistical language modeling," *ACM Trans. Asian Lang. Inf. Process.*, vol. 3, no. 2, pp. 87–93, 2004.

[6] R. Rosenfeld, "Two decades of statistical language modeling: Where do we go from here?" *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1270–1278, 2000.

[7] A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Seventh international conference on spoken language processing*, 2002.

[8] X. Liu and W. B. Croft, "Statistical language modeling for information retrieval," *Annu. Rev. Inf. Sci. Technol.*, vol. 39, no. 1, pp. 1–31, 2005.

[9] C. Zhai, *Statistical Language Models for Information Retrieval*, ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2008.

[10] S. M. Thede and M. P. Harper, "A second-order hidden markov model for part-of-speech tagging," in *27th Annual Meeting of the Association for Computational Linguistics, University of Maryland, College Park, Maryland, USA, 20-26 June 1999*, R. Dale and K. W. Church, Eds. ACL, 1999, pp. 175–182.

[11] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "A tree-based statistical language model for natural language speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 7, pp. 1001–1008, 1989.

[12] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, "Large language models in machine translation," in *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, J. Eisner, Ed. ACL, 2007, pp. 858–867.

[13] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 35, no. 3, pp. 400–401, 1987.

[14] W. A. Gale and G. Sampson, "Good-turing frequency estimation without tears," *J. Quant. Linguistics*, vol. 2, no. 3, pp. 217–237, 1995.

[15] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, 2003.

[16] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, T. Kobayashi, K. Hirose, and S. Nakamura, Eds. ISCA, 2010, pp. 1045–1048.

[17] S. Kombrink, T. Mikolov, M. Karafiát, and L. Burget, "Recurrent neural network based language modeling in meeting recognition," in *INTERSPEECH 2011, 12th Annual Conference of the International Speech Communication Association, Florence, Italy, August 27-31, 2011*. ISCA, 2011, pp. 2877–2880.

[18] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, 2011.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, Eds., 2013, pp. 3111–3119.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Ef-

ficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2013.

[21] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, M. A. Walker, H. Ji, and A. Stent, Eds. Association for Computational Linguistics, 2018, pp. 2227–2237.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017, pp. 5998–6008.

[23] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186.

[24] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 2020, pp. 7871–7880.

[25] W. Fedus, B. Zoph, and N. Shazeer, "Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity," *J. Mach. Learn. Res*, pp. 1–40, 2021.

[26] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, p. 9, 2019.

[27] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019.

[28] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, A. Raja, M. Dey, M. S. Bari, C. Xu, U. Thakker, S. S. Sharma, E. Szczechla, T. Kim, G. Chhablani, N. V. Nayak, D. Datta, J. Chang, M. T. Jiang, H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden, T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Févry, J. A. Fries, R. Teehan, T. L. Scao, S. Biderman, L. Gao, T. Wolf, and A. M. Rush, "Multitask prompted training enables zero-shot task generalization," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[29] T. Wang, A. Roberts, D. Hesslow, T. L. Scao, H. W. Chung, I. Beltagy, J. Launay, and C. Raffel, "What language model architecture and pretraining objective works best for zero-shot generalization?" in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ser. Proceedings of Machine Learning Research, vol. 162, 2022, pp. 22 964–22 984.

[30] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *CoRR*, vol. abs/2001.08361, 2020.

[31] M. Shanahan, "Talking about large language models," *CoRR*, vol. abs/2212.03551, 2022.

[32] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. H. Chi, Q. Le, and D. Zhou, "Chain of thought prompting elicits reasoning in large language models," *CoRR*, vol. abs/2201.11903, 2022.

[33] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, and L. Sifre, "Training compute-optimal large language models," vol. abs/2203.15556, 2022.

[34] R. Taylor, M. Kardas, G. Cucurull, T. Scialom, A. Hartshorn, E. Saravia, A. Poulton, V. Kerkez, and R. Stojnic, "Galactica: A large language model for science," *CoRR*, vol. abs/2211.09085, 2022.

[35] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Comput. Surv.*, pp. 195:1–195:35,

2023.

[36] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He, H. Peng, J. Li, J. Wu, Z. Liu, P. Xie, C. Xiong, J. Pei, P. S. Yu, and L. Sun, "A comprehensive survey on pretrained foundation models: A history from BERT to chatgpt," *CoRR*, vol. abs/2302.09419, 2023.

[37] X. Han, Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, Y. Yao, A. Zhang, L. Zhang, W. Han, M. Huang, Q. Jin, Y. Lan, Y. Liu, Z. Liu, Z. Lu, X. Qiu, R. Song, J. Tang, J. Wen, J. Yuan, W. X. Zhao, and J. Zhu, "Pre-trained models: Past, present and future," *AI Open*, vol. 2, pp. 225–250, 2021.

[38] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *CoRR*, vol. abs/2003.08271, 2020.

[39] S. Altman, "Planning for agi and beyond," *OpenAI Blog*, February 2023.

[40] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, H. Nori, H. Palangi, M. T. Ribeiro, and Y. Zhang, "Sparks of artificial general intelligence: Early experiments with gpt-4," vol. abs/2303.12712, 2023.

[41] S. Huang, L. Dong, W. Wang, Y. Hao, S. Singhal, S. Ma, T. Lv, L. Cui, O. K. Mohammed, B. Patra, Q. Liu, K. Aggarwal, Z. Chi, J. Bjorck, V. Chaudhary, S. Som, X. Song, and F. Wei, "Language is not all you need: Aligning perception with language models," *CoRR*, vol. abs/2302.14045, 2023.

[42] Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P. S. Yu, and L. Sun, "A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt," *arXiv preprint arXiv:2303.04226*, 2023.

[43] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu *et al.*, "Palm-e: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.

[44] C. Wu, S. Yin, W. Qi, X. Wang, Z. Tang, and N. Duan, "Visual chatgpt: Talking, drawing and editing with visual foundation models," *arXiv preprint arXiv:2303.04671*, 2023.

[45] OpenAI, "Gpt-4 technical report," *OpenAI*, 2023.

[46] Y. Fu, H. Peng, and T. Khot, "How does gpt obtain its ability? tracing emergent abilities of language models to their sources," *Yao Fu' s Notion*, Dec 2022.

[47] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus, "Emergent abilities of large language models," *CoRR*, vol. abs/2206.07682, 2022.

[48] J. Li, T. Tang, W. X. Zhao, and J. Wen, "Pretrained language model for text generation: A survey," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, Z. Zhou, Ed. ijcai.org, 2021, pp. 4492–4499.

[49] P. Lu, L. Qiu, W. Yu, S. Welleck, and K. Chang, "A survey of deep learning for mathematical reasoning," *CoRR*, vol. abs/2212.10535, 2022.

[50] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, L. Li, and Z. Sui, "A survey for in-context learning," *CoRR*, vol. abs/2301.00234, 2023.

[51] J. Huang and K. C. Chang, "Towards reasoning in large language models: A survey," *CoRR*, vol. abs/2212.10403, 2022.

[52] S. Qiao, Y. Ou, N. Zhang, X. Chen, Y. Yao, S. Deng, C. Tan, F. Huang, and H. Chen, "Reasoning with language model prompting: A survey," *CoRR*, vol. abs/2212.09597, 2022.

[53] J. Zhou, P. Ke, X. Qiu, M. Huang, and J. Zhang, "Chatgpt: potential, prospects, and limitations," in *Frontiers of Information Technology & Electronic Engineering*, 2023, pp. 1–6.

[54] W. X. Zhao, J. Liu, R. Ren, and J. Wen, "Dense text retrieval based on pretrained language models: A survey," *CoRR*, vol. abs/2211.14876, 2022.

[55] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020.

[56] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Is-

ard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, "Palm: Scaling language modeling with pathways," *CoRR*, vol. abs/2204.02311, 2022.

[57] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," *CoRR*, 2023.

[58] B. A. Huberman and T. Hogg, "Phase transitions in artificial intelligence systems," *Artificial Intelligence*, vol. 33, no. 2, pp. 155–171, 1987.

[59] J. W. Rae, S. Borgeaud, T. Cai, K. Millican, J. Hoffmann, H. F. Song, J. Aslanides, S. Henderson, R. Ring, S. Young, E. Rutherford, T. Hennigan, J. Menick, A. Cassirer, R. Powell, G. van den Driessche, L. A. Hendricks, M. Rauh, P. Huang, A. Glaese, J. Welbl, S. Dathathri, S. Huang, J. Uesato, J. Mellor, I. Higgins, A. Creswell, N. McAleese, A. Wu, E. Elsen, S. M. Jayakumar, E. Buchatskaya, D. Budden, E. Sutherland, K. Simonyan, M. Paganini, L. Sifre, L. Martens, X. L. Li, A. Kuncoro, A. Nematzadeh, E. Gribovskaya, D. Donato, A. Lazaridou, A. Mensch, J. Lespiau, M. Tsimpoukelli, N. Grigorev, D. Fritz, T. Sottiaux, M. Pajarskas, T. Pohlen, Z. Gong, D. Toyama, C. de Masson d'Autume, Y. Li, T. Terzi, V. Mikulik, I. Babuschkin, A. Clark, D. de Las Casas, A. Guy, C. Jones, J. Bradbury, M. J. Johnson, B. A. Hechtman, L. Weidinger, I. Gabriel, W. S. Isaac, E. Lockhart, S. Osindero, L. Rimell, C. Dyer, O. Vinyals, K. Ayoub, J. Stanway, L. Bennett, D. Hassabis, K. Kavukcuoglu, and G. Irving, "Scaling language models: Methods, analysis & insights from training gopher," *CoRR*, vol. abs/2112.11446, 2021.

[60] D. Dai, Y. Sun, L. Dong, Y. Hao, Z. Sui, and F. Wei, "Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers," *CoRR*, vol. abs/2212.10559, 2022.

[61] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," *CoRR*, vol. abs/2203.02155, 2022.

[62] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[63] A. Ananthaswamy, "In ai, is bigger always better?" *Nature*, 2023.

[64] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He, "Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters," in *KDD*, 2020, pp. 3505–3506.

[65] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, "Megatron-lm: Training multi-billion parameter language models using model parallelism," *CoRR*, vol. abs/1909.08053, 2019.

[66] D. Narayanan, M. Shoeybi, J. Casper, P. LeGresley, M. Patwary, V. Korthikanti, D. Vainbrand, P. Kashinkunti, J. Bernauer, B. Catanzaro, A. Phanishayee, and M. Zaharia, "Efficient large-scale language model training on GPU clusters using megatron-lm," in *International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2021, St. Louis, Missouri, USA, November 14-19, 2021*. ACM, 2021, p. 58.

[67] V. Korthikanti, J. Casper, S. Lym, L. McAfee, M. Andersch, M. Shoeybi, and B. Catanzaro, "Reducing activation recomputation in large transformer models," *CoRR*, vol. abs/2205.05198, 2022.

[68] T. L. Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilic, D. Hesslow, R. Castagné, A. S. Luccioni, F. Yvon, M. Gallé, J. Tow, A. M. Rush, S. Biderman, A. Webson, P. S. Ammanamanchi, T. Wang, B. Sagot, N. Muennighoff, A. V. del Moral, O. Ruwase, R. Bawden, S. Bekman, A. McMillan-Major, I. Beltagy, H. Nguyen, L. Saulnier, S. Tan, P. O. Suarez, V. Sanh, H. Laurençon, Y. Jernite, J. Launay, M. Mitchell, C. Raffel, A. Gokaslan, A. Simhi, A. Soroa, A. F. Aji, A. Alfassy, A. Rogers, A. K. Nitzav, C. Xu, C. Mou, C. Emezue, C. Klamm, C. Leong, D. van Strien, D. I. Adelani, and et al., "BLOOM: A 176b-parameter open-access multilingual language model," *CoRR*, vol. abs/2211.05100, 2022.

[69] P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 4299–4307.

[70] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom, "Toolformer: Language models can teach themselves to use tools," *CoRR*, vol. abs/2302.04761, 2023.

[71] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, X. Jiang, K. Cobbe, T. Eloundou, G. Krueger, K. Button, M. Knight, B. Chess, and J. Schulman, "Webgpt: Browser-assisted question-answering with human feedback," *CoRR*, vol. abs/2112.09332, 2021.

[72] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, pp. 140:1–140:67, 2020.

[73] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mt5: A massively multilingual pre-trained text-to-text transformer," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, 2021, pp. 483–498.

[74] W. Zeng, X. Ren, T. Su, H. Wang, Y. Liao, Z. Wang, X. Jiang, Z. Yang, K. Wang, X. Zhang, C. Li, Z. Gong, Y. Yao, X. Huang, J. Wang, J. Yu, Q. Guo, Y. Yu, Y. Zhang, J. Wang, H. Tao, D. Yan, Z. Yi, F. Peng, F. Jiang, H. Zhang, L. Deng, Y. Zhang, Z. Lin, C. Zhang, S. Zhang, M. Guo, S. Gu, G. Fan, Y. Wang, X. Jin, Q. Liu, and Y. Tian, "Pangu-α: Large-scale autoregressive pretrained chinese language models with auto-parallel computation," *CoRR*, vol. abs/2104.12369, 2021.

[75] Z. Zhang, Y. Gu, X. Han, S. Chen, C. Xiao, Z. Sun, Y. Yao, F. Qi, J. Guan, P. Ke, Y. Cai, G. Zeng, Z. Tan, Z. Liu, M. Huang, W. Han, Y. Liu, X. Zhu, and M. Sun, "CPM-2: large-scale cost-effective pre-trained language models," *CoRR*, vol. abs/2106.10715, 2021.

[76] E. Nijkamp, B. Pang, H. Hayashi, L. Tu, H. Wang, Y. Zhou, S. Savarese, and C. Xiong, "Codegen: An open large language model for code with mtulti-turn program synthesis," *arXiv preprint arXiv:2203.13474*, 2022.

[77] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang, M. Pieler, U. S. Prashanth, S. Purohit, L. Reynolds, J. Tow, B. Wang, and S. Weinbach, "Gpt-neox-20b: An open-source autoregressive language model," *CoRR*, vol. abs/2204.06745, 2022.

[78] Y. Wang, S. Mishra, P. Alipoormolabashi, Y. Kordi, A. Mirzaei, A. Naik, A. Ashok, A. S. Dhanasekaran, A. Arunkumar, D. Stap, E. Pathak, G. Karamanolakis, H. G. Lai, I. Purohit, I. Mondal, J. Anderson, K. Kuznia, K. Doshi, K. K. Pal, M. Patel, M. Moradshahi, M. Parmar, M. Purohit, N. Varshney, P. R. Kaza, P. Verma, R. S. Puri, R. Karia, S. Doshi, S. K. Sampat, S. Mishra, S. R. A, S. Patro, T. Dixit, and X. Shen, "Super-naturalinstructions: Generalization via declarative instructions on 1600+ NLP tasks," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 2022, pp. 5085–5109.

[79] Y. Tay, M. Dehghani, V. Q. Tran, X. García, J. Wei, X. Wang, H. W. Chung, D. Bahri, T. Schuster, H. Zheng, D. Zhou, N. Houlsby, and D. Metzler, "Ul2: Unifying language learning paradigms," 2022.

[80] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. T. Diab, X. Li, X. V. Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. S. Koura, A. Sridhar, T. Wang, and L. Zettlemoyer, "OPT: open pre-trained transformer language models," *CoRR*, vol. abs/2205.01068, 2022.

[81] M. R. Costa-jussà, J. Cross, O. Çelebi, M. Elbayad, K. Heafield, K. Heffernan, E. Kalbassi, J. Lam, D. Licht, J. Maillard, A. Sun, S. Wang, G. Wenzek, A. Youngblood, B. Akula, L. Barrault, G. M. Gonzalez, P. Hansanti, J. Hoffman, S. Jarrett, K. R. Sadagopan, D. Rowe, S. Spruit, C. Tran, P. Andrews, N. F. Ayan, S. Bhosale, S. Edunov, A. Fan, C. Gao, V. Goswami, F. Guzmán, P. Koehn, A. Mourachko, C. Ropers, S. Saleem, H. Schwenk, and J. Wang, "No language left behind: Scaling human-centered machine translation," *CoRR*, vol. abs/2207.04672, 2022.

[82] A. Zeng, X. Liu, Z. Du, Z. Wang, H. Lai, M. Ding, Z. Yang, Y. Xu, W. Zheng, X. Xia, W. L. Tam, Z. Ma,

Y. Xue, J. Zhai, W. Chen, P. Zhang, Y. Dong, and J. Tang, "GLM-130B: an open bilingual pre-trained model," vol. abs/2210.02414, 2022.

[83] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, S. Narang, G. Mishra, A. Yu, V. Y. Zhao, Y. Huang, A. M. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei, "Scaling instruction-finetuned language models," *CoRR*, vol. abs/2210.11416, 2022.

[84] N. Muennighoff, T. Wang, L. Sutawika, A. Roberts, S. Biderman, T. L. Scao, M. S. Bari, S. Shen, Z. X. Yong, H. Schoelkopf, X. Tang, D. Radev, A. F. Aji, K. Almubarak, S. Albanie, Z. Alyafeai, A. Webson, E. Raff, and C. Raffel, "Crosslingual generalization through multitask finetuning," *CoRR*, vol. abs/2211.01786, 2022.

[85] S. Iyer, X. V. Lin, R. Pasunuru, T. Mihaylov, D. Simig, P. Yu, K. Shuster, T. Wang, Q. Liu, P. S. Koura, X. Li, B. O'Horo, G. Pereyra, J. Wang, C. Dewan, A. Celikyilmaz, L. Zettlemoyer, and V. Stoyanov, "OPT-IML: scaling language model instruction meta learning through the lens of generalization," *CoRR*, vol. abs/2212.12017, 2022.

[86] S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff *et al.*, "Pythia: A suite for analyzing large language models across training and scaling," *arXiv preprint arXiv:2304.01373*, 2023.

[87] D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen, "Gshard: Scaling giant models with conditional computation and automatic sharding," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[88] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba, "Evaluating large language models trained on code," *CoRR*, vol. abs/2107.03374, 2021.

[89] Y. Sun, S. Wang, S. Feng, S. Ding, C. Pang, J. Shang, J. Liu, X. Chen, Y. Zhao, Y. Lu, W. Liu, Z. Wu, W. Gong, J. Liang, Z. Shang, P. Sun, W. Liu, X. Ouyang, D. Yu, H. Tian, H. Wu, and H. Wang, "ERNIE 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation," *CoRR*, vol. abs/2107.02137, 2021.

[90] O. Lieber, O. Sharir, B. Lenz, and Y. Shoham, "Jurassic-1: Technical details and evaluation," *White Paper. AI21 Labs*, vol. 1, 2021.

[91] B. Kim, H. Kim, S. Lee, G. Lee, D. Kwak, D. H. Jeon, S. Park, S. Kim, S. Kim, D. Seo, H. Lee, M. Jeong, S. Lee, M. Kim, S. Ko, S. Kim, T. Park, J. Kim, S. Kang, N. Ryu, K. M. Yoo, M. Chang, S. Suh, S. In, J. Park, K. Kim, H. Kim, J. Jeong, Y. G. Yeo, D. Ham, D. Park, M. Y. Lee, J. Kang, I. Kang, J. Ha, W. Park, and N. Sung, "What changes can large-scale language models bring? intensive study on hyperclova: Billions-scale korean generative pretrained transformers," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Association for Computational Linguistics, 2021.

[92] S. Wu, X. Zhao, T. Yu, R. Zhang, C. Shen, H. Liu, F. Li, H. Zhu, J. Luo, L. Xu *et al.*, "Yuan 1.0: Large-scale pre-trained language model in zero-shot and few-shot learning," *arXiv preprint arXiv:2110.04725*, 2021.

[93] A. Askell, Y. Bai, A. Chen, D. Drain, D. Ganguli, T. Henighan, A. Jones, N. Joseph, B. Mann, N. DasSarma, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, J. Kernion, K. Ndousse, C. Olsson, D. Amodei, T. B. Brown, J. Clark, S. McCandlish, C. Olah, and J. Kaplan, "A general language assistant as a laboratory for alignment," *CoRR*, vol. abs/2112.00861, 2021.

[94] S. Wang, Y. Sun, Y. Xiang, Z. Wu, S. Ding, W. Gong, S. Feng, J. Shang, Y. Zhao, C. Pang, J. Liu, X. Chen, Y. Lu, W. Liu, X. Wang, Y. Bai, Q. Chen, L. Zhao, S. Li, P. Sun, D. Yu, Y. Ma, H. Tian, H. Wu, T. Wu, W. Zeng, G. Li, W. Gao, and H. Wang, "ERNIE 3.0 titan: Exploring larger-scale knowledge enhanced pre-training for language understanding and generation," *CoRR*, vol. abs/2112.12731, 2021.

[95] N. Du, Y. Huang, A. M. Dai, S. Tong, D. Lepikhin, Y. Xu, M. Krikun, Y. Zhou, A. W. Yu, O. Firat, B. Zoph, L. Fedus, M. P. Bosma, Z. Zhou, T. Wang, Y. E. Wang, K. Webster, M. Pellat, K. Robinson, K. S. Meier-Hellstern, T. Duke, L. Dixon, K. Zhang, Q. V. Le, Y. Wu, Z. Chen, and C. Cui, "Glam: Efficient scaling of language models with mixture-of-experts," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, 2022, pp. 5547–5569.

[96] R. Thoppilan, D. D. Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H. Cheng, A. Jin, T. Bos, L. Baker, Y. Du, Y. Li, H. Lee, H. S. Zheng, A. Ghafouri, M. Menegali, Y. Huang, M. Krikun, D. Lepikhin, J. Qin, D. Chen, Y. Xu, Z. Chen, A. Roberts, M. Bosma, Y. Zhou, C. Chang, I. Krivokon, W. Rusch, M. Pickett, K. S. Meier-Hellstern, M. R. Morris, T. Doshi, R. D. Santos, T. Duke, J. Soraker, B. Zevenbergen, V. Prabhakaran, M. Diaz, B. Hutchinson, K. Olson, A. Molina, E. Hoffman-John, J. Lee, L. Aroyo, R. Rajakumar, A. Butryna, M. Lamm, V. Kuzmina, J. Fenton, A. Cohen, R. Bernstein, R. Kurzweil, B. Aguera-Arcas, C. Cui, M. Croak, E. H. Chi, and Q. Le, "Lamda: Language models for dialog applications," *CoRR*, vol. abs/2201.08239, 2022.

[97] S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhumoye, G. Zerveas, V. Korthikanti, E. Zheng, R. Child, R. Y. Aminabadi, J. Bernauer, X. Song, M. Shoeybi, Y. He, M. Houston, S. Tiwary, and B. Catanzaro, "Using deepspeed and megatron to train megatron-turing NLG 530b, A large-scale generative language model," *CoRR*, vol. abs/2201.11990, 2022.

[98] Y. Li, D. H. Choi, J. Chung, N. Kushman, J. Schrittwieser, R. Leblond, T. Eccles, J. Keeling, F. Gimeno, A. D. Lago, T. Hubert, P. Choy, C. de Masson d'Autume, I. Babuschkin, X. Chen, P. Huang, J. Welbl, S. Gowal, A. Cherepanov, J. Molloy, D. J. Mankowitz, E. S. Robson, P. Kohli, N. de Freitas, K. Kavukcuoglu, and O. Vinyals, "Competition-level code generation with alphacode," *Science*, 2022.

[99] S. Soltan, S. Ananthakrishnan, J. FitzGerald, R. Gupta, W. Hamza, H. Khan, C. Peris, S. Rawls, A. Rosenbaum, A. Rumshisky, C. S. Prakash, M. Sridhar, F. Triefenbach, A. Verma, G. Tür, and P. Natarajan, "Alexatm 20b: Few-shot learning using a large-scale multilingual seq2seq model," *CoRR*, vol. abs/2208.01448, 2022.

[100] A. Glaese, N. McAleese, M. Trebacz, J. Aslanides, V. Firoiu, T. Ewalds, M. Rauh, L. Weidinger, M. Chadwick, P. Thacker, L. Campbell-Gillingham, J. Uesato, P. Huang, R. Comanescu, F. Yang, A. See, S. Dathathri, R. Greig, C. Chen, D. Fritz, J. S. Elias, R. Green, S. Mokrá, N. Fernando, B. Wu, R. Foley, S. Young, I. Gabriel, W. Isaac, J. Mellor, D. Hassabis, K. Kavukcuoglu, L. A. Hendricks, and G. Irving, "Improving alignment of dialogue agents via targeted human judgements," *CoRR*, vol. abs/2209.14375, 2022.

[101] H. Su, X. Zhou, H. Yu, Y. Chen, Z. Zhu, Y. Yu, and J. Zhou, "Welm: A well-read pre-trained language model for chinese," *CoRR*, vol. abs/2209.10372, 2022.

[102] Y. Tay, J. Wei, H. W. Chung, V. Q. Tran, D. R. So, S. Shakeri, X. Garcia, H. S. Zheng, J. Rao, A. Chowdhery, D. Zhou, D. Metzler, S. Petrov, N. Houlsby, Q. V. Le, and M. Dehghani, "Transcending scaling laws with 0.1% extra compute," *CoRR*, vol. abs/2210.11399, 2022.

[103] X. Ren, P. Zhou, X. Meng, X. Huang, Y. Wang, W. Wang, P. Li, X. Zhang, A. Podolskiy, G. Arshinov, A. Bout, I. Piontkovskaya, J. Wei, X. Jiang, T. Su, Q. Liu, and J. Yao, "Pangu-Σ: Towards trillion parameter language model with sparse heterogeneous computing," *CoRR*, vol. abs/2303.10845, 2023.

[104] L. Huawei Technologies Co., "Huawei mindspore ai development framework," in *Artificial Intelligence Technology*. Springer, 2022, pp. 137–162.

[105] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Stanford alpaca: An instruction-following llama model," https://github.com/tatsu-lab/stanford_alpaca, 2023.

[106] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," 2023. [Online]. Available: https://vicuna.lmsys.org

[107] 2023. [Online]. Available: https://github.com/nebuly-ai/nebullvm/tree/main/apps/accelerate/chatllama

[108] Y. You, "Colossalchat: An open-source solution for cloning chatgpt with a complete rlhf pipeline," 2023. [Online]. Available: https://medium.com/@yangyou_berkeley/colossalchat-an-open-source-solution-for-cloning-chatgpt-with-a-complete-rlhf-pipeline-5edf08fb538b

[109] Y. Zhu, R. Kiros, R. S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," in *2015 IEEE*

International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015. IEEE Computer Society, 2015, pp. 19–27.

[110] "Project gutenberg." [Online]. Available: https://www.gutenberg.org/

[111] T. H. Trinh and Q. V. Le, "A simple method for commonsense reasoning," *CoRR*, vol. abs/1806.02847, 2018.

[112] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, "Defending against neural fake news," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 9051–9062.

[113] A. Gokaslan, V. C. E. Pavlick, and S. Tellex, "Openwebtext corpus," http://Skylion007.github.io/OpenWebTextCorpus, 2019.

[114] J. Baumgartner, S. Zannettou, B. Keegan, M. Squire, and J. Blackburn, "The pushshift reddit dataset," in *Proceedings of the Fourteenth International AAAI Conference on Web and Social Media, ICWSM 2020, Held Virtually, Original Venue: Atlanta, Georgia, USA, June 8-11, 2020*. AAAI Press, 2020, pp. 830–839.

[115] "Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/Main_Page

[116] "Bigquery dataset." [Online]. Available: https://cloud.google.com/bigquery?hl=zh-cn

[117] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, and C. Leahy, "The pile: An 800gb dataset of diverse text for language modeling," *CoRR*, vol. abs/2101.00027, 2021.

[118] H. Laurençon, L. Saulnier, T. Wang, C. Akiki, A. V. del Moral, T. Le Scao, L. Von Werra, C. Mou, E. G. Ponferrada, H. Nguyen *et al.*, "The bigscience roots corpus: A 1.6 tb composite multilingual dataset," in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[119] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[120] "Common crawl." [Online]. Available: https://commoncrawl.org/

[121] "A reproduction version of cc-stories on hugging face." [Online]. Available: https://huggingface.co/datasets/spacemanidol/cc-stories

[122] B. Wang and A. Komatsuzaki, "GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model," https://github.com/kingoflolz/mesh-transformer-jax, 2021.

[123] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*. Association for Computational Linguistics, 2020, pp. 38–45.

[124] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018. [Online]. Available: http://github.com/google/jax

[125] Z. Bian, H. Liu, B. Wang, H. Huang, Y. Li, C. Wang, F. Cui, and Y. You, "Colossal-ai: A unified deep learning system for large-scale parallel training," *CoRR*, vol. abs/2110.14883, 2021.

[126] J. Fang, Y. Yu, S. Li, Y. You, and J. Zhou, "Patrickstar: Parallel training of pre-trained models via a chunk-based memory management," *CoRR*, vol. abs/2108.05818, 2021.

[127] "Bmtrain: Effient training for big models." [Online]. Available: https://github.com/OpenBMB/BMTrain

[128] J. He, J. Qiu, A. Zeng, Z. Yang, J. Zhai, and J. Tang, "Fastmoe: A fast mixture-of-expert training system," *CoRR*, vol. abs/2103.13262, 2021.

[129] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 8024–8035.

[130] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis,

J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. A. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, November 2-4, 2016*, K. Keeton and T. Roscoe, Eds. USENIX Association, 2016, pp. 265–283.

[131] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," *CoRR*, vol. abs/1512.01274, 2015.

[132] Y. Ma, D. Yu, T. Wu, and H. Wang, "Paddlepaddle: An open-source deep learning platform from industrial practice," *Frontiers of Data and Domputing*, vol. 1, no. 1, p. 105, 2019.

[133] J. Yuan, X. Li, C. Cheng, J. Liu, R. Guo, S. Cai, C. Yao, F. Yang, X. Yi, C. Wu, H. Zhang, and J. Zhao, "Oneflow: Redesign the distributed deep learning framework from scratch," *CoRR*, vol. abs/2110.15032, 2021.

[134] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, E. M. Smith, Y. Boureau, and J. Weston, "Recipes for building an open-domain chatbot," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, 2021, pp. 300–325.

[135] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, Y. Wu, B. Neyshabur, G. Gur-Ari, and V. Misra, "Solving quantitative reasoning problems with language models," *CoRR*, vol. abs/2206.14858, 2022.

[136] T. Saier, J. Krause, and M. Färber, "unarxive 2022: All arxiv publications pre-processed for nlp, including structured full-text and citation network," *arXiv preprint arXiv:2303.14957*, 2023.

[137] H. A. Simon, "Experiments with a heuristic compiler," *J. ACM*, vol. 10, no. 4, pp. 493–506, 1963.

[138] Z. Manna and R. J. Waldinger, "Toward automatic program synthesis," *Commun. ACM*, vol. 14, no. 3, pp. 151–165, 1971.

[139] Z. Feng, D. Guo, D. Tang, N. Duan, X. Feng, M. Gong, L. Shou, B. Qin, T. Liu, D. Jiang, and M. Zhou, "Codebert: A pre-trained model for programming and natural languages," in *Findings of EMNLP*, 2020.

[140] J. Austin, A. Odena, M. I. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. J. Cai, M. Terry, Q. V. Le, and C. Sutton, "Program synthesis with large language models," *CoRR*, vol. abs/2108.07732, 2021.

[141] S. Black, L. Gao, P. Wang, C. Leahy, and S. Biderman, "GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow," 2021.

[142] F. F. Xu, U. Alon, G. Neubig, and V. J. Hellendoorn, "A systematic evaluation of large language models of code," in *MAPS@PLDI*, 2022.

[143] D. Fried, A. Aghajanyan, J. Lin, S. Wang, E. Wallace, F. Shi, R. Zhong, W. Yih, L. Zettlemoyer, and M. Lewis, "Incoder: A generative model for code infilling and synthesis," in *ICLR*, 2023.

[144] A. Madaan, S. Zhou, U. Alon, Y. Yang, and G. Neubig, "Language models of code are few-shot commonsense learners," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Association for Computational Linguistics, 2022, pp. 1384–1403.

[145] Y. Wu, A. Q. Jiang, W. Li, M. N. Rabe, C. Staats, M. Jamnik, and C. Szegedy, "Autoformalization with large language models," *CoRR*, vol. abs/2205.12615, 2022.

[146] D. Hernandez, T. B. Brown, T. Conerly, N. DasSarma, D. Drain, S. E. Showk, N. Elhage, Z. Hatfield-Dodds, T. Henighan, T. Hume, S. Johnston, B. Mann, C. Olah, C. Olsson, D. Amodei, N. Joseph, J. Kaplan, and S. McCandlish, "Scaling laws and interpretability of learning from repeated data," *CoRR*, vol. abs/2205.10487, 2022.

[147] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[148] K. Lee, D. Ippolito, A. Nystrom, C. Zhang, D. Eck, C. Callison-Burch, and N. Carlini, "Deduplicating training data makes language models better," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 2022, pp. 8424–8445.

[149] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramèr, and C. Zhang, "Quantifying memorization across neural

language models," *CoRR*, 2022.

[150] N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. B. Brown, D. Song, Ú. Erlingsson, A. Oprea, and C. Raffel, "Extracting training data from large language models," in *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, 2021, pp. 2633–2650.

[151] N. Kandpal, E. Wallace, and C. Raffel, "Deduplicating training data mitigates privacy risks in language models," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*. PMLR, 2022, pp. 10 697–10 707.

[152] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, E. Blanco and W. Lu, Eds. Association for Computational Linguistics, 2018.

[153] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.

[154] M. Davis and M. Dürst, "Unicode normalization forms," 2001.

[155] D. Paperno, G. Kruszewski, A. Lazaridou, Q. N. Pham, R. Bernardi, S. Pezzelle, M. Baroni, G. Boleda, and R. Fernández, "The LAMBADA dataset: Word prediction requiring a broad discourse context," in *ACL (1)*. The Association for Computer Linguistics, 2016.

[156] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, "Deep double descent: Where bigger models and more data hurt," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[157] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H. Hon, "Unified language model pre-training for natural language understanding and generation," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019, pp. 13 042–13 054.

[158] A. Clark, D. de Las Casas, A. Guy, A. Mensch, M. Paganini, J. Hoffmann, B. Damoc, B. A. Hechtman, T. Cai, S. Borgeaud, G. van den Driessche, E. Rutherford, T. Hennigan, M. J. Johnson, A. Cassirer, C. Jones, E. Buchatskaya, D. Budden, L. Sifre, S. Osindero, O. Vinyals, M. Ranzato, J. W. Rae, E. Elsen, K. Kavukcuoglu, and K. Simonyan, "Unified scaling laws for routed language models," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, 2022, pp. 4057–4086.

[159] L. J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," vol. abs/1607.06450, 2016.

[160] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu, "On layer normalization in the transformer architecture," in *ICML*, 2020.

[161] M. Ding, Z. Yang, W. Hong, W. Zheng, C. Zhou, D. Yin, J. Lin, X. Zou, Z. Shao, H. Yang, and J. Tang, "Cogview: Mastering text-to-image generation via transformers," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021, pp. 19 822–19 835.

[162] B. Zhang and R. Sennrich, "Root mean square layer normalization," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019, pp. 12 360–12 371.

[163] H. Wang, S. Ma, L. Dong, S. Huang, D. Zhang, and F. Wei, "Deepnet: Scaling transformers to 1, 000 layers," vol. abs/2203.00555, 2022.

[164] T. L. Scao, T. Wang, D. Hesslow, S. Bekman, M. S. Bari, S. Biderman, H. Elsahar, N. Muennighoff, J. Phang, O. Press, C. Raffel, V. Sanh, S. Shen, L. Sutawika, J. Tae, Z. X. Yong, J. Launay, and I. Beltagy, "What language model to train if you have one million GPU hours?" in *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 2022, pp. 765–782.

[165] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[166] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017, pp. 933–941.

[167] N. Shazeer, "GLU variants improve transformer," vol. abs/2002.05202, 2020.

[168] S. Narang, H. W. Chung, Y. Tay, L. Fedus, T. Févry, M. Matena, K. Malkan, N. Fiedel, N. Shazeer, Z. Lan, Y. Zhou, W. Li, N. Ding, J. Marcus, A. Roberts, and C. Raffel, "Do transformer modifications transfer across implementations and applications?" in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, 2021, pp. 5758–5773.

[169] O. Press, N. A. Smith, and M. Lewis, "Train short, test long: Attention with linear biases enables input length extrapolation," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

[170] J. Su, Y. Lu, S. Pan, B. Wen, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," vol. abs/2104.09864, 2021.

[171] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers," *CoRR*, vol. abs/1904.10509, 2019.

[172] H. Peng, N. Pappas, D. Yogatama, R. Schwartz, N. A. Smith, and L. Kong, "Random feature attention," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

[173] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontañón, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed, "Big bird: Transformers for longer sequences," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[174] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Re, "Flashattention: Fast and memory-efficient exact attention with IO-awareness," in *NeurIPS*, 2022.

[175] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[176] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *CoRR*, vol. abs/1711.05101, 2017.

[177] N. Shazeer and M. Stern, "Adafactor: Adaptive learning rates with sublinear memory cost," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 4603–4611.

[178] Y. Huang, Y. Cheng, A. Bapna, O. Firat, D. Chen, M. X. Chen, H. Lee, J. Ngiam, Q. V. Le, Y. Wu, and Z. Chen, "Gpipe: Efficient training of giant neural networks using pipeline parallelism," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 103–112.

[179] A. Harlap, D. Narayanan, A. Phanishayee, V. Seshadri, N. R. Devanur, G. R. Ganger, and P. B. Gibbons, "Pipedream: Fast and efficient pipeline parallel DNN training," *CoRR*, vol. abs/1806.03377, 2018.

[180] S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He, "Zero: memory optimizations toward training trillion parameter models," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2020, Virtual Event / Atlanta, Georgia, USA, November 9-19, 2020*, C. Cuicchi, I. Qualters, and W. T. Kramer, Eds. IEEE/ACM, 2020, p. 20.

[181] P. Micikevicius, S. Narang, J. Alben, G. F. Diamos, E. Elsen, D. García, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu, "Mixed precision training," *CoRR*, vol. abs/1710.03740, 2017.

[182] Q. Xu, S. Li, C. Gong, and Y. You, "An efficient 2d method for training super-large deep learning models," *CoRR*, vol. abs/2104.05343, 2021.

[183] B. Wang, Q. Xu, Z. Bian, and Y. You, "Tesseract: Parallelize the tensor parallelism efficiently," in *Proceedings of the 51st International Conference on Parallel Processing, ICPP 2022, Bordeaux, France, 29 August 2022 - 1 September 2022*. ACM, 2022.

[184] Z. Bian, Q. Xu, B. Wang, and Y. You, "Maximizing parallelism in distributed training for huge neural networks," *CoRR*, vol. abs/2105.14450, 2021.

[185] S. Li, F. Xue, C. Baranwal, Y. Li, and Y. You, "Sequence parallelism: Long sequence training from system perspective," *arXiv e-prints*, pp. arXiv–2105, 2021.

[186] FairScale authors, "Fairscale: A general purpose modular pytorch library for high performance and large scale training," https://github.com/facebookresearch/fairscale, 2021.

[187] L. Zheng, Z. Li, H. Zhang, Y. Zhuang, Z. Chen, Y. Huang, Y. Wang, Y. Xu, D. Zhuo, E. P. Xing *et al.*, "Alpa: Automating inter-and {Intra-Operator} parallelism for distributed deep learning," in *OSDI*, 2022, pp. 559–578.

[188] T. Chen, B. Xu, C. Zhang, and C. Guestrin, "Training deep nets with sublinear memory cost," *CoRR*, vol. abs/1604.06174, 2016.

[189] Z. Yao, C. Li, X. Wu, S. Youn, and Y. He, "A comprehensive study on post-training quantization for large language models," *CoRR*, vol. abs/2303.08302, 2023.

[190] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "Llm.int8(): 8-bit matrix multiplication for transformers at scale," *CoRR*, vol. abs/2208.07339, 2022.

[191] C. Tao, L. Hou, W. Zhang, L. Shang, X. Jiang, Q. Liu, P. Luo, and N. Wong, "Compression of generative pre-trained language models via quantization," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Association for Computational Linguistics, 2022, pp. 4821–4836.

[192] S. Mishra, D. Khashabi, C. Baral, and H. Hajishirzi, "Cross-task generalization via natural language crowdsourcing instructions," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., 2022, pp. 3470–3487.

[193] Q. Ye, B. Y. Lin, and X. Ren, "Crossfit: A few-shot learning challenge for cross-task generalization in NLP," in *EMNLP (1)*. Association for Computational Linguistics, 2021, pp. 7163–7189.

[194] S. H. Bach, V. Sanh, Z. X. Yong, A. Webson, C. Raffel, N. V. Nayak, A. Sharma, T. Kim, M. S. Bari, T. Févry, Z. Alyafeai, M. Dey, A. Santilli, Z. Sun, S. Ben-David, C. Xu, G. Chhablani, H. Wang, J. A. Fries, M. S. AlShaibani, S. Sharma, U. Thakker, K. Almubarak, X. Tang, D. R. Radev, M. T. Jiang, and A. M. Rush, "Promptsource: An integrated development environment and repository for natural language prompts," in *ACL (demo)*. Association for Computational Linguistics, 2022, pp. 93–104.

[195] V. Aribandi, Y. Tay, T. Schuster, J. Rao, H. S. Zheng, S. V. Mehta, H. Zhuang, V. Q. Tran, D. Bahri, J. Ni, J. P. Gupta, K. Hui, S. Ruder, and D. Metzler, "Ext5: Towards extreme multi-task scaling for transfer learning," in *ICLR*. OpenReview.net, 2022.

[196] T. Xie, C. H. Wu, P. Shi, R. Zhong, T. Scholak, M. Yasunaga, C. Wu, M. Zhong, P. Yin, S. I. Wang, V. Zhong, B. Wang, C. Li, C. Boyle, A. Ni, Z. Yao, D. Radev, C. Xiong, L. Kong, R. Zhang, N. A. Smith, L. Zettlemoyer, and T. Yu, "Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models," in *EMNLP*. Association for Computational Linguistics, 2022, pp. 602–631.

[197] T. Tang, J. Li, W. X. Zhao, and J. Wen, "MVP: multi-task supervised pre-training for natural language generation," *CoRR*, vol. abs/2206.12131, 2022.

[198] R. Lou, K. Zhang, and W. Yin, "Is prompt all you need? no. A comprehensive and broader view of instruction learning," *CoRR*, vol. abs/2303.10475, 2023.

[199] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," in *ACL (1)*. Association for Computational Linguistics, 2019, pp. 4487–4496.

[200] A. Aghajanyan, A. Gupta, A. Shrivastava, X. Chen, L. Zettlemoyer, and S. Gupta, "Muppet: Massive multi-task representations with pre-finetuning," in *EMNLP (1)*. Association for Computational Linguistics, 2021, pp. 5799–5811.

[201] S. Longpre, L. Hou, T. Vu, A. Webson, H. W. Chung, Y. Tay, D. Zhou, Q. V. Le, B. Zoph, J. Wei, and A. Roberts, "The flan collection: Designing data and methods for effective instruction tuning," *CoRR*, vol. abs/2301.13688, 2023.

[202] Y. Gu, P. Ke, X. Zhu, and M. Huang, "Learning instructions with unlabeled data for zero-shot cross-task generalization," in *EMNLP*. Association for Computational Linguistics, 2022, pp. 1617–1634.

[203] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, "Self-instruct: Aligning language model with self generated instructions," *CoRR*, vol. abs/2212.10560, 2022.

[204] O. Honovich, T. Scialom, O. Levy, and T. Schick, "Unnatural instructions: Tuning language models with (almost) no human labor," *CoRR*, vol. abs/2212.09689, 2022.

[205] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Stanford alpaca: An instruction-following llama model," https://github.com/tatsu-lab/stanford_alpaca, 2023.

[206] Z. Kenton, T. Everitt, L. Weidinger, I. Gabriel, V. Mikulik, and G. Irving, "Alignment of language

agents," *CoRR*, vol. abs/2103.14659, 2021.

[207] A. Askell, Y. Bai, A. Chen, D. Drain, D. Ganguli, T. Henighan, A. Jones, N. Joseph, B. Mann, N. DasSarma, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, J. Kernion, K. Ndousse, C. Olsson, D. Amodei, T. B. Brown, J. Clark, S. McCandlish, C. Olah, and J. Kaplan, "A general language assistant as a laboratory for alignment," *CoRR*, vol. abs/2112.00861, 2021.

[208] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, N. Joseph, S. Kadavath, J. Kernion, T. Conerly, S. E. Showk, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, T. Hume, S. Johnston, S. Kravec, L. Lovitt, N. Nanda, C. Olsson, D. Amodei, T. B. Brown, J. Clark, S. McCandlish, C. Olah, B. Mann, and J. Kaplan, "Training a helpful and harmless assistant with reinforcement learning from human feedback," *CoRR*, vol. abs/2204.05862, 2022.

[209] E. Perez, S. Huang, H. F. Song, T. Cai, R. Ring, J. Aslanides, A. Glaese, N. McAleese, and G. Irving, "Red teaming language models with language models," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Association for Computational Linguistics, 2022, pp. 3419–3448.

[210] D. Ganguli, L. Lovitt, J. Kernion, A. Askell, Y. Bai, S. Kadavath, B. Mann, E. Perez, N. Schiefer, K. Ndousse, A. Jones, S. Bowman, A. Chen, T. Conerly, N. DasSarma, D. Drain, N. Elhage, S. E. Showk, S. Fort, Z. Hatfield-Dodds, T. Henighan, D. Hernandez, T. Hume, J. Jacobson, S. Johnston, S. Kravec, C. Olsson, S. Ringer, E. Tran-Johnson, D. Amodei, T. Brown, N. Joseph, S. McCandlish, C. Olah, J. Kaplan, and J. Clark, "Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned," *CoRR*, vol. abs/2209.07858, 2022.

[211] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. F. Christiano, and G. Irving, "Fine-tuning language models from human preferences," *CoRR*, vol. abs/1909.08593, 2019.

[212] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano, "Learning to summarize from human feedback," *CoRR*, vol. abs/2009.01325, 2020.

[213] J. Menick, M. Trebacz, V. Mikulik, J. Aslanides, H. F. Song, M. Chadwick, M. Glaese, S. Young, L. Campbell-Gillingham, G. Irving, and N. McAleese, "Teaching language models to support answers with verified quotes," *CoRR*, vol. abs/2203.11147, 2022.

[214] J. Wu, L. Ouyang, D. M. Ziegler, N. Stiennon, R. Lowe, J. Leike, and P. F. Christiano, "Recursively summarizing books with human feedback," *CoRR*, vol. abs/2109.10862, 2021.

[215] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[216] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer, "Rethinking the role of demonstrations: What makes in-context learning work?" in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*. Association for Computational Linguistics, 2022, pp. 11 048–11 064.

[217] Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp, "Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds., 2022, pp. 8086–8098.

[218] Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh, "Calibrate before use: Improving few-shot performance of language models," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, M. Meila and T. Zhang, Eds., 2021, pp. 12 697–12 706.

[219] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What makes good in-context examples for gpt-3?" in *Proceedings of Deep Learning Inside Out: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, DeeLIO@ACL 2022, Dublin, Ireland and Online, May 27, 2022*, 2022, pp. 100–114.

[220] Y. Lee, C. Lim, and H. Choi, "Does GPT-3 generate empathetic dialogues? A novel in-context example selection method and automatic evaluation metric for empathetic dialogue generation," in *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, N. Calzolari, C. Huang, H. Kim,

J. Pustejovsky, L. Wanner, K. Choi, P. Ryu, H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T. K. Lee, E. Santus, F. Bond, and S. Na, Eds. International Committee on Computational Linguistics, 2022, pp. 669–683.

[221] I. Levy, B. Bogin, and J. Berant, "Diverse demonstrations improve in-context compositional generalization," *CoRR*, vol. abs/2212.06800, 2022.

[222] H. Su, J. Kasai, C. H. Wu, W. Shi, T. Wang, J. Xin, R. Zhang, M. Ostendorf, L. Zettlemoyer, N. A. Smith, and T. Yu, "Selective annotation makes language models better few-shot learners," *CoRR*, 2022.

[223] X. Ye, S. Iyer, A. Celikyilmaz, V. Stoyanov, G. Durrett, and R. Pasunuru, "Complementary explanations for effective in-context learning," *CoRR*, 2022.

[224] X. Li and X. Qiu, "Finding supporting examples for in-context learning," *CoRR*, 2023.

[225] O. Rubin, J. Herzig, and J. Berant, "Learning to retrieve prompts for in-context learning," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, 2022, pp. 2655–2671.

[226] Y. Zhang, S. Feng, and C. Tan, "Active example selection for in-context learning," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 2022, pp. 9134–9148.

[227] F. Gilardi, M. Alizadeh, and M. Kubli, "Chatgpt outperforms crowd-workers for text-annotation tasks," 2023.

[228] H. J. Kim, H. Cho, J. Kim, T. Kim, K. M. Yoo, and S. Lee, "Self-generated in-context learning: Leveraging auto-regressive language models as a demonstration generator," *CoRR*, vol. abs/2206.08082, 2022.

[229] Y. Lin, A. Papangelis, S. Kim, S. Lee, D. Hazarika, M. Namazifar, D. Jin, Y. Liu, and D. Hakkani-Tur, "Selective in-context data augmentation for intent detection using pointwise v-information," *CoRR*, 2023.

[230] S. M. Xie, A. Raghunathan, P. Liang, and T. Ma, "An explanation of in-context learning as implicit bayesian inference," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

[231] Z. Zhang, A. Zhang, M. Li, and A. Smola, "Automatic chain of thought prompting in large language models," *CoRR*, vol. abs/2210.03493, 2022.

[232] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, O. Bousquet, Q. Le, and E. H. Chi, "Least-to-most prompting enables complex reasoning in large language models," *CoRR*, vol. abs/2205.10625, 2022.

[233] Z. Wu, Y. Wang, J. Ye, and L. Kong, "Self-adaptive in-context learning," *CoRR*, vol. abs/2212.10375, 2022.

[234] S. Min, M. Lewis, L. Zettlemoyer, and H. Hajishirzi, "Metaicl: Learning to learn in context," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, M. Carpuat, M. de Marneffe, and I. V. M. Ruíz, Eds., 2022, pp. 2791–2809.

[235] S. C. Y. Chan, A. Santoro, A. K. Lampinen, J. X. Wang, A. Singh, P. H. Richemond, J. McClelland, and F. Hill, "Data distributional properties drive emergent in-context learning in transformers," *CoRR*, vol. abs/2205.05055, 2022.

[236] S. Shin, S. Lee, H. Ahn, S. Kim, H. Kim, B. Kim, K. Cho, G. Lee, W. Park, J. Ha, and N. Sung, "On the effect of pretraining corpora on in-context learning by a large-scale language model," in *NAACL-HLT*. Association for Computational Linguistics, 2022, pp. 5168–5186.

[237] J. von Oswald, E. Niklasson, E. Randazzo, J. Sacramento, A. Mordvintsev, A. Zhmoginov, and M. Vladymyrov, "Transformers learn in-context by gradient descent," *CoRR*, vol. abs/2212.07677, 2022.

[238] C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, S. Johnston, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah, "In-context learning and induction heads," *CoRR*, vol. abs/2209.11895, 2022.

[239] H. Bansal, K. Gopalakrishnan, S. Dingliwal, S. Bodapati, K. Kirchhoff, and D. Roth, "Rethinking the role of scale for in-context learning: An interpretability-based case study at 66 billion scale," *CoRR*, vol. abs/2212.09095, 2022.

[240] Y. Li, M. E. Ildiz, D. S. Papailiopoulos, and S. Oymak, "Transformers as algorithms: Generalization and implicit model selection in in-context learning," *CoRR*, vol. abs/2301.07067, 2023.

[241] E. Akyürek, D. Schuurmans, J. Andreas, T. Ma, and D. Zhou, "What learning algorithm is in-context learning? investigations with linear models," *CoRR*, vol. abs/2211.15661, 2022.

[242] S. Garg, D. Tsipras, P. Liang, and G. Valiant, "What can transformers learn in-context? A case study of simple function classes," *CoRR*, vol. abs/2208.01066, 2022.

[243] K. Cobbe, V. Kosaraju, M. Bavarian, J. Hilton, R. Nakano, C. Hesse, and J. Schulman, "Training verifiers to solve math word problems," *CoRR*, vol. abs/2110.14168, 2021.

[244] A. Patel, S. Bhattamishra, and N. Goyal, "Are NLP models really able to solve simple math word problems?" in *NAACL-HLT*. Association for Computational Linguistics, 2021, pp. 2080–2094.

[245] S. Miao, C. Liang, and K. Su, "A diverse corpus for evaluating and developing english math word problem solvers," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 975–984.

[246] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4149–4158.

[247] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, and J. Berant, "Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies," *Trans. Assoc. Comput. Linguistics*, vol. 9, pp. 346–361, 2021.

[248] Y. Li, Z. Lin, S. Zhang, Q. Fu, B. Chen, J. Lou, and W. Chen, "On the advance of making language models better reasoners," *CoRR*, vol. abs/2206.02336, 2022.

[249] Y. Fu, H. Peng, A. Sabharwal, P. Clark, and T. Khot, "Complexity-based prompting for multi-step reasoning," *CoRR*, vol. abs/2210.00720, 2022.

[250] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *CoRR*, vol. abs/2205.11916, 2022.

[251] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," *CoRR*, vol. abs/2203.11171, 2022.

[252] ——, "Rationale-augmented ensembles in language models," *CoRR*, 2022.

[253] S. Imani, L. Du, and H. Shrivastava, "Mathprompter: Mathematical reasoning using large language models," *arXiv preprint arXiv:2303.05398*, 2023.

[254] E. Zelikman, J. Mu, N. D. Goodman, and Y. T. Wu, "Star: Self-taught reasoner bootstrapping reasoning with reasoning," 2022.

[255] J. Huang, S. S. Gu, L. Hou, Y. Wu, X. Wang, H. Yu, and J. Han, "Large language models can self-improve," *CoRR*, vol. abs/2210.11610, 2022.

[256] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, T. Linzen, G. Chrupala, and A. Alishahi, Eds. Association for Computational Linguistics, 2018, pp. 353–355.

[257] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, B. Newman, B. Yuan, B. Yan, C. Zhang, C. Cosgrove, C. D. Manning, C. Ré, D. Acosta-Navas, D. A. Hudson, E. Zelikman, E. Durmus, F. Ladhak, F. Rong, H. Ren, H. Yao, J. Wang, K. Santhanam, L. J. Orr, L. Zheng, M. Yüksekgönül, M. Suzgun, N. Kim, N. Guha, N. S. Chatterji, O. Khattab, P. Henderson, Q. Huang, R. Chi, S. M. Xie, S. Santurkar, S. Ganguli, T. Hashimoto, T. Icard, T. Zhang, V. Chaudhary, W. Wang, X. Li, Y. Mai, Y. Zhang, and Y. Koreeda, "Holistic evaluation of language models," *CoRR*, vol. abs/2211.09110, 2022.

[258] A. Madaan and A. Yazdanbakhsh, "Text and patterns: For effective chain of thought, it takes two to tango," *CoRR*, vol. abs/2209.07686, 2022.

[259] Z. Zhang, A. Zhang, M. Li, H. Zhao, G. Karypis, and A. Smola, "Multimodal chain-of-thought reasoning in language models," *CoRR*, vol. abs/2302.00923, 2023.

[260] F. Shi, M. Suzgun, M. Freitag, X. Wang, S. Srivats, S. Vosoughi, H. W. Chung, Y. Tay, S. Ruder, D. Zhou, D. Das, and J. Wei, "Language models are multilingual chain-of-thought reasoners," *CoRR*, vol. abs/2210.03057, 2022.

[261] K. Shridhar, A. Stolfo, and M. Sachan, "Distilling multi-step reasoning capabilities of large language models into smaller models via semantic decompositions," *ArXiv*, vol. abs/2212.00193, 2022.

[262] N. Ho, L. Schmid, and S. Yun, "Large language models are reasoning teachers," *CoRR*, vol. abs/2212.10071, 2022.

[263] L. C. Magister, J. Mallinson, J. Adámek, E. Malmi, and A. Severyn, "Teaching small language models to reason," *CoRR*, vol. abs/2212.08410, 2022.

[264] Y. Fu, H. Peng, L. Ou, A. Sabharwal, and T. Khot, "Specializing smaller language models towards multi-step reasoning," *CoRR*, vol. abs/2301.12726, 2023.

[265] A. Chan, Z. Zeng, W. Lake, B. Joshi, H. Chen, and X. Ren, "Knife: Distilling meta-reasoning knowledge with free-text rationales," in *ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML*.

[266] Z. Li, C. Wang, P. Ma, C. Liu, S. Wang, D. Wu, and C. Gao, "On the feasibility of specialized ability stealing for large language code models," *CoRR*, 2023.

[267] Z. Dai, V. Y. Zhao, J. Ma, Y. Luan, J. Ni, J. Lu, A. Bakalov, K. Guu, K. B. Hall, and M. Chang, "Promptagator: Few-shot dense retrieval from 8 examples," *CoRR*, 2022.

[268] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of english: The penn treebank," *Comput. Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[269] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," in *ICLR (Poster)*. OpenReview.net, 2017.

[270] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, R. Soricut, L. Specia, and A. Tamchyna, "Findings of the 2014 workshop on statistical machine translation," in *WMT@ACL*. The Association for Computer Linguistics, 2014, pp. 12–58.

[271] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, M. Huck, A. Jimeno-Yepes, P. Koehn, V. Logacheva, C. Monz, M. Negri, A. Névéol, M. L. Neves, M. Popel, M. Post, R. Rubino, C. Scarton, L. Specia, M. Turchi, K. Verspoor, and M. Zampieri, "Findings of the 2016 conference on machine translation," in *WMT*. The Association for Computer Linguistics, 2016, pp. 131–198.

[272] L. Barrault, O. Bojar, M. R. Costa-jussà, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, P. Koehn, S. Malmasi, C. Monz, M. Müller, S. Pal, M. Post, and M. Zampieri, "Findings of the 2019 conference on machine translation (WMT19)," in *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, O. Bojar, R. Chatterjee, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, A. Jimeno-Yepes, P. Koehn, A. Martins, C. Monz, M. Negri, A. Névéol, M. L. Neves, M. Post, M. Turchi, and K. Verspoor, Eds. Association for Computational Linguistics, 2019, pp. 1–61.

[273] L. Barrault, M. Biesialska, O. Bojar, M. R. Costa-jussà, C. Federmann, Y. Graham, R. Grundkiewicz, B. Haddow, M. Huck, E. Joanis, T. Kocmi, P. Koehn, C. Lo, N. Ljubesic, C. Monz, M. Morishita, M. Nagata, T. Nakazawa, S. Pal, M. Post, and M. Zampieri, "Findings of the 2020 conference on machine translation (WMT20)," in *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, L. Barrault, O. Bojar, F. Bougares, R. Chatterjee, M. R. Costa-jussà, C. Federmann, M. Fishel, A. Fraser, Y. Graham, P. Guzman, B. Haddow, M. Huck, A. Jimeno-Yepes, P. Koehn, A. Martins, M. Morishita, C. Monz, M. Nagata, T. Nakazawa, and M. Negri, Eds. Association for Computational Linguistics, 2020, pp. 1–55.

[274] F. Akhbardeh, A. Arkhangorodsky, M. Biesialska, O. Bojar, R. Chatterjee, V. Chaudhary, M. R. Costa-jussà, C. España-Bonet, A. Fan, C. Federmann, M. Freitag, Y. Graham, R. Grundkiewicz, B. Haddow, L. Harter, K. Heafield, C. Homan, M. Huck, K. Amponsah-Kaakyire, J. Kasai, D. Khashabi, K. Knight, T. Kocmi, P. Koehn, N. Lourie, C. Monz, M. Morishita, M. Nagata, A. Nagesh, T. Nakazawa, M. Negri, S. Pal, A. A. Tapo, M. Turchi, V. Vydrin, and M. Zampieri, "Findings of the 2021 conference on machine translation (WMT21)," in *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, L. Barrault, O. Bojar, F. Bougares, R. Chatterjee, M. R. Costa-jussà, C. Federmann, M. Fishel, A. Fraser, M. Freitag, Y. Graham, R. Grundkiewicz, P. Guzman, B. Haddow, M. Huck, A. Jimeno-Yepes, P. Koehn, T. Kocmi, A. Martins, M. Morishita, and C. Monz, Eds. Association for Computational Linguistics, 2021, pp. 1–88.

[275] T. Kocmi, R. Bawden, O. Bojar, A. Dvorkovich, C. Fe-

dermann, M. Fishel, T. Gowda, Y. Graham, R. Grundkiewicz, B. Haddow, R. Knowles, P. Koehn, C. Monz, M. Morishita, M. Nagata, T. Nakazawa, M. Novák, M. Popel, and M. Popovic, "Findings of the 2022 conference on machine translation (WMT22)," in *Proceedings of the Seventh Conference on Machine Translation, WMT 2022, Abu Dhabi, United Arab Emirates (Hybrid), December 7-8, 2022*, P. Koehn, L. Barrault, O. Bojar, F. Bougares, R. Chatterjee, M. R. Costa-jussà, C. Federmann, M. Fishel, A. Fraser, M. Freitag, Y. Graham, R. Grundkiewicz, P. Guzman, B. Haddow, M. Huck, A. Jimeno-Yepes, T. Kocmi, A. Martins, M. Morishita, C. Monz, M. Nagata, T. Nakazawa, M. Negri, A. Névéol, M. Neves, M. Popel, M. Turchi, and M. Zampieri, Eds. Association for Computational Linguistics, 2022, pp. 1–45.

[276] N. Goyal, C. Gao, V. Chaudhary, P. Chen, G. Wenzek, D. Ju, S. Krishnan, M. Ranzato, F. Guzmán, and A. Fan, "The flores-101 evaluation benchmark for low-resource and multilingual machine translation," *Trans. Assoc. Comput. Linguistics*, vol. 10, pp. 522–538, 2022.

[277] R. Bawden, E. Bilinski, T. Lavergne, and S. Rosset, "Diabla: a corpus of bilingual spontaneous written dialogues for machine translation," *Lang. Resour. Evaluation*, vol. 55, no. 3, pp. 635–660, 2021.

[278] R. Nallapati, B. Zhou, C. N. dos Santos, Ç. Gülçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence rnns and beyond," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, Y. Goldberg and S. Riezler, Eds. ACL, 2016, pp. 280–290.

[279] S. Narayan, S. B. Cohen, and M. Lapata, "Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization," in *EMNLP*. Association for Computational Linguistics, 2018, pp. 1797–1807.

[280] F. Ladhak, E. Durmus, C. Cardie, and K. Mckeown, "Wikilingua: A new benchmark dataset for cross-lingual abstractive summarization," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020, pp. 4034–4048.

[281] S. Moon, P. Shah, A. Kumar, and R. Subba, "Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs," in *ACL (1)*. Association for Computational Linguistics, 2019, pp. 845–854.

[282] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Superglue: A stickier benchmark for general-purpose language understanding systems," in *NeurIPS*, 2019, pp. 3261–3275.

[283] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, "Measuring massive multitask language understanding," in *ICLR*. OpenReview.net, 2021.

[284] M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, A. Chowdhery, Q. V. Le, E. H. Chi, D. Zhou, and J. Wei, "Challenging big-bench tasks and whether chain-of-thought can solve them," *CoRR*, vol. abs/2210.09261, 2022.

[285] L. Xu, H. Hu, X. Zhang, L. Li, C. Cao, Y. Li, Y. Xu, K. Sun, D. Yu, C. Yu, Y. Tian, Q. Dong, W. Liu, B. Shi, Y. Cui, J. Li, J. Zeng, R. Wang, W. Xie, Y. Li, Y. Patterson, Z. Tian, Y. Zhang, H. Zhou, S. Liu, Z. Zhao, Q. Zhao, C. Yue, X. Zhang, Z. Yang, K. Richardson, and Z. Lan, "CLUE: A chinese language understanding evaluation benchmark," in *COLING*. International Committee on Computational Linguistics, 2020, pp. 4762–4772.

[286] D. Hendrycks, S. Basart, S. Kadavath, M. Mazeika, A. Arora, E. Guo, C. Burns, S. Puranik, H. He, D. Song, and J. Steinhardt, "Measuring coding challenge competence with APPS," in *NeurIPS Datasets and Benchmarks*, 2021.

[287] Y. Lai, C. Li, Y. Wang, T. Zhang, R. Zhong, L. Zettlemoyer, S. W. Yih, D. Fried, S. I. Wang, and T. Yu, "DS-1000: A natural and reliable benchmark for data science code generation," *CoRR*, vol. abs/2211.11501, 2022.

[288] Z. Wang, S. Zhou, D. Fried, and G. Neubig, "Execution-based evaluation for open-domain code generation," *CoRR*, vol. abs/2212.10481, 2022.

[289] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. P. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M. Chang, A. M. Dai, J. Uszkoreit, Q. Le, and S. Petrov, "Natural questions: a benchmark for question answering research," *Trans. Assoc. Comput. Linguistics*, pp. 452–466, 2019.

[290] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord, "Think you have solved question answering? try arc, the AI2 reasoning challenge," *CoRR*, vol. abs/1803.05457, 2018.

[291] S. Lin, J. Hilton, and O. Evans, "Truthfulqa: Measuring

how models mimic human falsehoods," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 2022, pp. 3214–3252.

[292] J. Berant, A. Chou, R. Frostig, and P. Liang, "Semantic parsing on freebase from question-answer pairs," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2013, pp. 1533–1544.

[293] M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer, "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 2017, pp. 1601–1611.

[294] Y. Bisk, R. Zellers, R. L. Bras, J. Gao, and Y. Choi, "PIQA: reasoning about physical commonsense in natural language," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 2020, pp. 7432–7439.

[295] M. Dubey, D. Banerjee, A. Abdelkawi, and J. Lehmann, "Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia," in *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, 2019, pp. 69–78.

[296] Y. Gu, S. Kase, M. Vanni, B. M. Sadler, P. Liang, X. Yan, and Y. Su, "Beyond I.I.D.: three levels of generalization for question answering on knowledge bases," in *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, 2021, pp. 3477–3488.

[297] S. Cao, J. Shi, L. Pan, L. Nie, Y. Xiang, L. Hou, J. Li, B. He, and H. Zhang, "KQA pro: A dataset with explicit compositional programs for complex question answering over knowledge base," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, 2022, pp. 6101–6119.

[298] X. Hu, X. Wu, Y. Shu, and Y. Qu, "Logical form generation via multi-task learning for complex question answering over knowledge bases," in *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, 2022, pp. 1687–1696.

[299] S. Longpre, Y. Lu, and J. Daiber, "MKQA: A linguistically diverse benchmark for multilingual open domain question answering," *Trans. Assoc. Comput. Linguistics*, vol. 9, pp. 1389–1406, 2021.

[300] T. Saikh, T. Ghosal, A. Mittal, A. Ekbal, and P. Bhattacharyya, "Scienceqa: a novel resource for question answering on scholarly articles," *Int. J. Digit. Libr.*, vol. 23, no. 3, pp. 289–301, 2022.

[301] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal, "Can a suit of armor conduct electricity? A new dataset for open book question answering," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 2018, pp. 2381–2391.

[302] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, "MS MARCO: A human generated machine reading comprehension dataset," in *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, 2016.

[303] T. Khot, P. Clark, M. Guerquin, P. Jansen, and A. Sabharwal, "QASC: A dataset for question answering via sentence composition," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 2020, pp. 8082–8090.

[304] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100, 000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, 2016, pp. 2383–2392.

[305] A. H. Miller, A. Fisch, J. Dodge, A. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language*

Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, 2016, pp. 1400–1409.

[306] B. Goodrich, V. Rao, P. J. Liu, and M. Saleh, "Assessing the factual accuracy of generated text," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, 2019, pp. 166–175.

[307] K. Toutanova and D. Chen, "Observed versus latent features for knowledge base and text inference," in *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality, CVSC 2015, Beijing, China, July 26-31, 2015*, 2015, pp. 57–66.

[308] K. D. Bollacker, C. Evans, P. K. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10-12, 2008*, 2008, pp. 1247–1250.

[309] T. Dettmers, P. Minervini, P. Stenetorp, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018, pp. 1811–1818.

[310] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, pp. 39–41, 1995.

[311] F. Petroni, T. Rocktäschel, S. Riedel, P. S. H. Lewis, A. Bakhtin, Y. Wu, and A. H. Miller, "Language models as knowledge bases?" in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 2019, pp. 2463–2473.

[312] F. Mahdisoltani, J. Biega, and F. M. Suchanek, "YAGO3: A knowledge base from multilingual wikipedias," in *Seventh Biennial Conference on Innovative Data Systems Research, CIDR 2015, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*, 2015.

[313] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, 2007, pp. 697–706.

[314] C. Clark, K. Lee, M. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova, "Boolq: Exploring the surprising difficulty of natural yes/no questions," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 2924–2936.

[315] M. Sap, H. Rashkin, D. Chen, R. L. Bras, and Y. Choi, "Socialiqa: Commonsense reasoning about social interactions," *CoRR*, vol. abs/1904.09728, 2019.

[316] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "Hellaswag: Can a machine really finish your sentence?" in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, A. Korhonen, D. R. Traum, and L. Màrquez, Eds. Association for Computational Linguistics, 2019, pp. 4791–4800.

[317] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi, "Winogrande: An adversarial winograd schema challenge at scale," in *AAAI*. AAAI Press, 2020, pp. 8732–8740.

[318] M. Roemmele, C. A. Bejan, and A. S. Gordon, "Choice of plausible alternatives: An evaluation of commonsense causal reasoning," in *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21-23, 2011*. AAAI, 2011.

[319] K. Sakaguchi, C. Bhagavatula, R. L. Bras, N. Tandon, P. Clark, and Y. Choi, "proscript: Partially ordered scripts generation," in *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, M. Moens, X. Huang, L. Specia, and S. W. Yih, Eds. Association for Computational Linguistics, 2021, pp. 2138–2149.

[320] B. Dalvi, L. Huang, N. Tandon, W. Yih, and P. Clark, "Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6,*

2018, Volume 1 (Long Papers), M. A. Walker, H. Ji, and A. Stent, Eds. Association for Computational Linguistics, 2018, pp. 1595–1604.

[321] S. Saha, P. Yadav, L. Bauer, and M. Bansal, "Explagraphs: An explanation graph generation task for structured commonsense reasoning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, M. Moens, X. Huang, L. Specia, and S. W. Yih, Eds. Association for Computational Linguistics, 2021, pp. 7716–7740.

[322] O. Tafjord, B. Dalvi, and P. Clark, "Proofwriter: Generating implications, proofs, and abductive statements over natural language," in *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, ser. Findings of ACL, C. Zong, F. Xia, W. Li, and R. Navigli, Eds., vol. ACL/IJCNLP 2021. Association for Computational Linguistics, 2021, pp. 3621–3634.

[323] B. Dalvi, P. Jansen, O. Tafjord, Z. Xie, H. Smith, L. Pipatanangkura, and P. Clark, "Explaining answers with entailment trees," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, M. Moens, X. Huang, L. Specia, and S. W. Yih, Eds. Association for Computational Linguistics, 2021, pp. 7358–7370.

[324] A. Saparov and H. He, "Language models are greedy reasoners: A systematic formal analysis of chain-of-thought," *CoRR*, vol. abs/2210.01240, 2022.

[325] C. Anil, Y. Wu, A. Andreassen, A. Lewkowycz, V. Misra, V. V. Ramasesh, A. Slone, G. Gur-Ari, E. Dyer, and B. Neyshabur, "Exploring length generalization in large language models," *CoRR*, vol. abs/2207.04901, 2022.

[326] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, A. Kluska, A. Lewkowycz, A. Agarwal, A. Power, A. Ray, A. Warstadt, A. W. Kocurek, A. Safaya, A. Tazarv, A. Xiang, A. Parrish, A. Nie, A. Hussain, A. Askell, A. Dsouza, A. Rahane, A. S. Iyer, A. Andreassen, A. Santilli, A. Stuhlmüller, A. M. Dai, A. La, A. K. Lampinen, A. Zou, A. Jiang, A. Chen, A. Vuong, A. Gupta, A. Gottardi, A. Norelli, A. Venkatesh, A. Gholamidavoodi, A. Tabassum, A. Menezes, A. Kirubarajan, A. Mullokandov, A. Sabharwal, A. Herrick, A. Efrat, A. Erdem, A. Karakas, and et al., "Beyond the imitation game: Quantifying and extrapolating the capabilities of language models," *CoRR*, vol. abs/2206.04615, 2022.

[327] L. Gao, A. Madaan, S. Zhou, U. Alon, P. Liu, Y. Yang, J. Callan, and G. Neubig, "PAL: program-aided language models," *CoRR*, vol. abs/2211.10435, 2022.

[328] S. Roy and D. Roth, "Solving general arithmetic word problems," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, Eds. The Association for Computational Linguistics, 2015, pp. 1743–1752.

[329] A. Amini, S. Gabriel, S. Lin, R. Koncel-Kedziorski, Y. Choi, and H. Hajishirzi, "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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 2357–2367.

[330] W. Ling, D. Yogatama, C. Dyer, and P. Blunsom, "Program induction by rationale generation: Learning to solve and explain algebraic word problems," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, R. Barzilay and M. Kan, Eds. Association for Computational Linguistics, 2017, pp. 158–167.

[331] R. Koncel-Kedziorski, S. Roy, A. Amini, N. Kushman, and H. Hajishirzi, "Mawps: A math word problem repository," in *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1152–1157.

[332] D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner, "DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 2019, pp. 2368–2378.

[333] S. Welleck, J. Liu, R. L. Bras, H. Hajishirzi, Y. Choi,

and K. Cho, "Naturalproofs: Mathematical theorem proving in natural language," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, J. Vanschoren and S. Yeung, Eds., 2021.

[334] A. Q. Jiang, W. Li, J. M. Han, and Y. Wu, "Lisa: Language models of isabelle proofs," in *6th Conference on Artificial Intelligence and Theorem Proving*, 2021, pp. 378–392.

[335] K. Zheng, J. M. Han, and S. Polu, "minif2f: a cross-system benchmark for formal olympiad-level mathematics," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[336] Z. Azerbayev, B. Piotrowski, H. Schoelkopf, E. W. Ayers, D. Radev, and J. Avigad, "Proofnet: Autoformalizing and formally proving undergraduate-level mathematics," *CoRR*, vol. abs/2302.12433, 2023.

[337] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.

[338] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *EMNLP*. The Association for Computational Linguistics, 2015, pp. 379–389.

[339] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading wikipedia to answer open-domain questions," in *ACL (1)*. Association for Computational Linguistics, 2017, pp. 1870–1879.

[340] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*. ACL, 2002, pp. 311–318.

[341] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Association for Computational Linguistics, Jul. 2004, pp. 74–81.

[342] K. Yang, Y. Tian, N. Peng, and D. Klein, "Re3: Generating longer stories with recursive reprompting and revision," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Association for Computational Linguistics, 2022, pp. 4393–4479.

[343] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, Q. V. Do, Y. Xu, and P. Fung, "A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity," *CoRR*, vol. abs/2302.04023, 2023.

[344] S. Gulwani, O. Polozov, and R. Singh, "Program synthesis," *Found. Trends Program. Lang.*, vol. 4, no. 1-2, pp. 1–119, 2017.

[345] S. Zhang, Z. Chen, Y. Shen, M. Ding, J. B. Tenenbaum, and C. Gan, "Planning with large language models for code generation," 2023.

[346] M. Welsh, "The end of programming," *Commun. ACM*, vol. 66, no. 1, pp. 34–35, 2023.

[347] B. Wang, X. Deng, and H. Sun, "Iteratively prompt pre-trained language models for chain of thought," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Association for Computational Linguistics, 2022, pp. 2714–2730.

[348] O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis, "Measuring and narrowing the compositionality gap in language models," *CoRR*, vol. abs/2210.03350, 2022.

[349] J. Ye, X. Chen, N. Xu, C. Zu, Z. Shao, S. Liu, Y. Cui, Z. Zhou, C. Gong, Y. Shen, J. Zhou, S. Chen, T. Gui, Q. Zhang, and X. Huang, "A comprehensive capability analysis of gpt-3 and gpt-3.5 series models," *arXiv preprint arXiv:2303.10420*, 2023.

[350] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, 1989, pp. 109–165.

[351] R. Kemker, M. McClure, A. Abitino, T. L. Hayes, and C. Kanan, "Measuring catastrophic forgetting in neural networks," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018, pp. 3390–3398.

[352] A. Roberts, C. Raffel, and N. Shazeer, "How much knowledge can you pack into the parameters of a language model?" in *Proceedings of the 2020 Conference*

on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, 2020, pp. 5418–5426.

[353] G. Izacard, P. S. H. Lewis, M. Lomeli, L. Hosseini, F. Petroni, T. Schick, J. Dwivedi-Yu, A. Joulin, S. Riedel, and E. Grave, "Few-shot learning with retrieval augmented language models," *CoRR*, vol. abs/2208.03299, 2022.

[354] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, 2020, pp. 3929–3938.

[355] P. S. H. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[356] Y. Lan, G. He, J. Jiang, J. Jiang, W. X. Zhao, and J. Wen, "Complex knowledge base question answering: A survey," *CoRR*, vol. abs/2108.06688, 2021.

[357] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. van den Driessche, J. Lespiau, B. Damoc, A. Clark, D. de Las Casas, A. Guy, J. Menick, R. Ring, T. Hennigan, S. Huang, L. Maggiore, C. Jones, A. Cassirer, A. Brock, M. Paganini, G. Irving, O. Vinyals, S. Osindero, K. Simonyan, J. W. Rae, E. Elsen, and L. Sifre, "Improving language models by retrieving from trillions of tokens," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, 2022, pp. 2206–2240.

[358] B. Peng, M. Galley, P. He, H. Cheng, Y. Xie, Y. Hu, Q. Huang, L. Liden, Z. Yu, W. Chen, and J. Gao, "Check your facts and try again: Improving large language models with external knowledge and automated feedback," *CoRR*, vol. abs/2302.12813, 2023.

[359] S. Agarwal, I. Akkaya, V. Balcom, M. Bavarian, G. Bernadett-Shapiro, G. Brockman, M. Brundage, J. Chan, F. Chantzis, N. Deutsch, B. Eastman, A. Eleti, N. Felix, S. P. Fishman, I. Fulford, C. Gibson, J. Gross, M. Heaton, J. Hilton, X. Hu, S. Jain, H. Jin, L. Kilpatrick, C. Kim, M. Kolhede, A. Mayne, P. McMillan, D. Medina, J. Menick, A. Mishchenko, A. Nair, R. Nayak, A. Neelakantan, R. Nuttall, J. Parish, A. T. Passos, A. Perelman, F. de Avila Belbute Peres, V. Pong, J. Schulman, E. Sigler, N. Staudacher, N. Turley, J. Tworek, R. Greene, A. Vijayvergiya, C. Voss, J. Weng, M. Wiethoff, S. Yoo, K. Yu, W. Zaremba, S. Zhao, W. Zhuk, and B. Zoph, "Chatgpt plugins," *OpenAI Blog*, March 2023.

[360] A. Lazaridou, E. Gribovskaya, W. Stokowiec, and N. Grigorev, "Internet-augmented language models through few-shot prompting for open-domain question answering," *CoRR*, vol. abs/2203.05115, 2022.

[361] A. Madaan, N. Tandon, P. Clark, and Y. Yang, "Memory-assisted prompt editing to improve GPT-3 after deployment," in *EMNLP*. Association for Computational Linguistics, 2022, pp. 2833–2861.

[362] D. Dai, L. Dong, Y. Hao, Z. Sui, B. Chang, and F. Wei, "Knowledge neurons in pretrained transformers," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Association for Computational Linguistics, 2022, pp. 8493–8502.

[363] K. Meng, D. Bau, A. J. Andonian, and Y. Belinkov, "Locating and editing factual associations in gpt," in *Advances in Neural Information Processing Systems*, 2022.

[364] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, and W. Chen, "Synthetic prompting: Generating chain-of-thought demonstrations for large language models," *CoRR*, vol. abs/2302.00618, 2023.

[365] N. Bian, X. Han, L. Sun, H. Lin, Y. Lu, and B. He, "ChatGPT is a Knowledgeable but Inexperienced Solver: An Investigation of Commonsense Problem in Large Language Models," *CoRR*, 2023.

[366] M. I. Nye, A. J. Andreassen, G. Gur-Ari, H. Michalewski, J. Austin, D. Bieber, D. Dohan, A. Lewkowycz, M. Bosma, D. Luan, C. Sutton, and A. Odena, "Show your work: Scratchpads for intermediate computation with language models," *CoRR*, vol. abs/2112.00114, 2021.

[367] J. Qian, H. Wang, Z. Li, S. Li, and X. Yan, "Limitations of language models in arithmetic and symbolic induction," *CoRR*, vol. abs/2208.05051, 2022.

[368] W. X. Zhao, K. Zhou, Z. Gong, B. Zhang, Y. Zhou,

J. Sha, Z. Chen, S. Wang, C. Liu, and J. Wen, "Jiuzhang: A chinese pre-trained language model for mathematical problem understanding," in *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, A. Zhang and H. Rangwala, Eds. ACM, 2022, pp. 4571–4581.

[369] Q. Wang, C. Kaliszyk, and J. Urban, "First experiments with neural translation of informal to formal mathematics," in *Intelligent Computer Mathematics - 11th International Conference, CICM 2018, Hagenberg, Austria, August 13-17, 2018, Proceedings*, ser. Lecture Notes in Computer Science, F. Rabe, W. M. Farmer, G. O. Passmore, and A. Youssef, Eds., vol. 11006. Springer, 2018, pp. 255–270.

[370] S. Polu and I. Sutskever, "Generative language modeling for automated theorem proving," *CoRR*, vol. abs/2009.03393, 2020.

[371] A. Q. Jiang, W. Li, S. Tworkowski, K. Czechowski, T. Odrzygózdz, P. Milos, Y. Wu, and M. Jamnik, "Thor: Wielding hammers to integrate language models and automated theorem provers," *CoRR*, vol. abs/2205.10893, 2022.

[372] S. Polu, J. M. Han, K. Zheng, M. Baksys, I. Babuschkin, and I. Sutskever, "Formal mathematics statement curriculum learning," *CoRR*, vol. abs/2202.01344, 2022.

[373] A. Q. Jiang, S. Welleck, J. P. Zhou, W. Li, J. Liu, M. Jamnik, T. Lacroix, Y. Wu, and G. Lample, "Draft, sketch, and prove: Guiding formal theorem provers with informal proofs," *CoRR*, vol. abs/2210.12283, 2022.

[374] Q. Lyu, S. Havaldar, A. Stein, L. Zhang, D. Rao, E. Wong, M. Apidianaki, and C. Callison-Burch, "Faithful chain-of-thought reasoning," *CoRR*, vol. abs/2301.13379, 2023.

[375] Y. Weng, M. Zhu, S. He, K. Liu, and J. Zhao, "Large language models are reasoners with self-verification," *CoRR*, vol. abs/2212.09561, 2022.

[376] X. Pi, Q. Liu, B. Chen, M. Ziyadi, Z. Lin, Q. Fu, Y. Gao, J. Lou, and W. Chen, "Reasoning like program executors," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 2022, pp. 761–779.

[377] A. Parisi, Y. Zhao, and N. Fiedel, "TALM: tool augmented language models," *CoRR*, vol. abs/2205.12255, 2022.

[378] N. Nangia, C. Vania, R. Bhalerao, and S. R. Bowman, "Crows-pairs: A challenge dataset for measuring social biases in masked language models," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 2020, pp. 1953–1967.

[379] R. Rudinger, J. Naradowsky, B. Leonard, and B. V. Durme, "Gender bias in coreference resolution," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, 2018, pp. 8–14.

[380] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 2022, pp. 9118–9147.

[381] T. Carta, C. Romac, T. Wolf, S. Lamprier, O. Sigaud, and P. Oudeyer, "Grounding large language models in interactive environments with online reinforcement learning," *CoRR*, vol. abs/2302.02662, 2023.

[382] X. Puig, K. Ra, M. Boben, J. Li, T. Wang, S. Fidler, and A. Torralba, "Virtualhome: Simulating household activities via programs," in *CVPR*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 8494–8502.

[383] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "ALFRED: A benchmark for interpreting grounded instructions for everyday tasks," in *CVPR*. Computer Vision Foundation / IEEE, 2020, pp. 10 737–10 746.

[384] S. Srivastava, C. Li, M. Lingelbach, R. Martín-Martín, F. Xia, K. E. Vainio, Z. Lian, C. Gokmen, S. Buch, C. K. Liu, S. Savarese, H. Gweon, J. Wu, and L. Fei-Fei, "BEHAVIOR: benchmark for everyday household activities in virtual, interactive, and ecological environments," in *CoRL*, ser. Proceedings of Machine Learning Research, vol. 164. PMLR, 2021, pp. 477–490.

[385] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, and M. Yan, "Do as

I can, not as I say: Grounding language in robotic affordances," *CoRR*, vol. abs/2204.01691, 2022.

[386] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," *CoRR*, vol. abs/2209.07753, 2022.

[387] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," *CoRR*, vol. abs/2209.11302, 2022.

[388] J. H. Clark, J. Palomaki, V. Nikolaev, E. Choi, D. Garrette, M. Collins, and T. Kwiatkowski, "Tydi QA: A benchmark for information-seeking question answering in typologically diverse languages," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 454–470, 2020.

[389] L. Gao, J. Tow, S. Biderman, S. Black, A. DiPofi, C. Foster, L. Golding, J. Hsu, K. McDonell, N. Muennighoff, J. Phang, L. Reynolds, E. Tang, A. Thite, B. Wang, K. Wang, and A. Zou, "A framework for few-shot language model evaluation," Sep. 2021.

[390] Q. Zhong, L. Ding, J. Liu, B. Du, and D. Tao, "Can chatgpt understand too? A comparative study on chatgpt and fine-tuned BERT," *CoRR*, vol. abs/2302.10198, 2023.

[391] J. Kocon, I. Cichecki, O. Kaszyca, M. Kochanek, D. Szydlo, J. Baran, J. Bielaniewicz, M. Gruza, A. Janz, K. Kanclerz, A. Kocon, B. Koptyra, W. Mieleszczenko-Kowszewicz, P. Milkowski, M. Oleksy, M. Piasecki, L. Radlinski, K. Wojtasik, S. Wozniak, and P. Kazienko, "Chatgpt: Jack of all trades, master of none," *CoRR*, vol. abs/2302.10724, 2023.

[392] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, and D. Yang, "Is chatgpt a general-purpose natural language processing task solver?" *CoRR*, vol. abs/2302.06476, 2023.

[393] Y. Ma, Y. Cao, Y. Hong, and A. Sun, "Large language model is not a good few-shot information extractor, but a good reranker for hard samples!" *CoRR*, vol. abs/2303.08559, 2023.

[394] X. Chen, J. Ye, C. Zu, N. Xu, R. Zheng, M. Peng, J. Zhou, T. Gui, Q. Zhang, and X. Huang, "How robust is gpt-3.5 to predecessors? a comprehensive study on language understanding tasks," 2023.

[395] M. Jang and T. Lukasiewicz, "Consistency analysis of chatgpt," *CoRR*, vol. abs/2303.06273, 2023.

[396] R. Tang, X. Han, X. Jiang, and X. Hu, "Does synthetic data generation of llms help clinical text mining?" *arXiv preprint arXiv:2303.04360*, 2023.

[397] O. Nov, N. Singh, and D. M. Mann, "Putting chatgpt's medical advice to the (turing) test," *CoRR*, vol. abs/2301.10035, 2023.

[398] S. Chen, B. H. Kann, M. B. Foote, H. J. Aerts, G. K. Savova, R. H. Mak, and D. S. Bitterman, "The utility of chatgpt for cancer treatment information," *medRxiv*, 2023.

[399] L. Yunxiang, L. Zihan, Z. Kai, D. Ruilong, and Z. You, "Chatdoctor: A medical chat model fine-tuned on llama model using medical domain knowledge," 2023.

[400] K. Jeblick, B. Schachtner, J. Dexl, A. Mittermeier, A. T. Stüber, J. Topalis, T. Weber, P. Wesp, B. O. Sabel, J. Ricke, and M. Ingrisch, "Chatgpt makes medicine easy to swallow: An exploratory case study on simplified radiology reports," *CoRR*, vol. abs/2212.14882, 2022.

[401] H. Nori, N. King, S. M. McKinney, D. Carignan, and E. Horvitz, "Capabilities of gpt-4 on medical challenge problems," vol. abs/2303.13375, 2023.

[402] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, and Y. Wu, "How close is chatgpt to human experts? comparison corpus, evaluation, and detection," *CoRR*, vol. abs/2301.07597, 2023.

[403] V. Liévin, C. E. Hother, and O. Winther, "Can large language models reason about medical questions?" *CoRR*, vol. abs/2207.08143, 2022.

[404] G. Kortemeyer, "Could an artificial-intelligence agent pass an introductory physics course?" *arXiv preprint arXiv:2301.12127*, 2023.

[405] S. Bordt and U. von Luxburg, "Chatgpt participates in a computer science exam," *CoRR*, vol. abs/2303.09461, 2023.

[406] K. Malinka, M. Peresíni, A. Firc, O. Hujnak, and F. Janus, "On the educational impact of chatgpt: Is artificial intelligence ready to obtain a university degree?" *CoRR*, vol. abs/2303.11146, 2023.

[407] T. Susnjak, "Chatgpt: The end of online exam integrity?" *CoRR*, vol. abs/2212.09292, 2022.

[408] A. Blair-Stanek, N. Holzenberger, and B. V. Durme, "Can GPT-3 perform statutory reasoning?" *CoRR*, vol. abs/2302.06100, 2023.

[409] F. Yu, L. Quartey, and F. Schilder, "Legal prompting: Teaching a language model to think like a lawyer," *CoRR*, vol. abs/2212.01326, 2022.

[410] D. Trautmann, A. Petrova, and F. Schilder, "Legal prompt engineering for multilingual legal judgement

prediction," *CoRR*, vol. abs/2212.02199, 2022.

[411] J. H. Choi, K. E. Hickman, A. Monahan, and D. Schwarcz, "Chatgpt goes to law school," *Available at SSRN*, 2023.

[412] J. J. Nay, "Law informs code: A legal informatics approach to aligning artificial intelligence with humans," *CoRR*, vol. abs/2209.13020, 2022.

[413] A. Tamkin, M. Brundage, J. Clark, and D. Ganguli, "Understanding the capabilities, limitations, and societal impact of large language models," *CoRR*, vol. abs/2102.02503, 2021.

[414] Z. Sun, "A short survey of viewing large language models in legal aspect," *CoRR*, vol. abs/2303.09136, 2023.

[415] A. Abid, M. Farooqi, and J. Zou, "Persistent anti-muslim bias in large language models," in *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19-21, 2021*, M. Fourcade, B. Kuipers, S. Lazar, and D. K. Mulligan, Eds. ACM, 2021, pp. 298–306.

[416] A. Borji, "A categorical archive of chatgpt failures," *CoRR*, vol. abs/2302.03494, 2023.

[417] M. Kosinski, "Theory of mind may have spontaneously emerged in large language models," *CoRR*, vol. abs/2302.02083, 2023.

[418] M. M. Amin, E. Cambria, and B. W. Schuller, "Will affective computing emerge from foundation models and general ai? A first evaluation on chatgpt," *CoRR*, vol. abs/2303.03186, 2023.

[419] R. Aiyappa, J. An, H. Kwak, and Y.-Y. Ahn, "Can we trust the evaluation on chatgpt?" vol. abs/2303.12767, 2023.

[420] H. Cho, H. J. Kim, J. Kim, S. Lee, S. Lee, K. M. Yoo, and T. Kim, "Prompt-augmented linear probing: Scaling beyond the limit of few-shot in-context learners," *CoRR*, vol. abs/2212.10873, 2022.

[421] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," *ACM Comput. Surv.*, vol. 55, no. 6, pp. 109:1–109:28, 2023.