机器之心报道

编辑：张倩、小舟

GPT-3
对一些问题的回答令人大跌眼镜，但它可能只是想要一句「鼓励」。

「一个玩杂耍的人总共有 16 个球，其中一半是高尔夫球，高尔夫球中又有一半是蓝色的球，请问蓝球总共有多少个？」

对于一个小学生来说，这是一道再简单不过的数学题。但看似无所不能的 GPT-3 却被这道题难住了。

如果你输入这个问题之后，直接向 GPT-3 发问：「问题的答案（阿拉伯数字）是：__？」它会「不假思索」地给出一个错误答案：8。

GPT-3：你别说准不准，你就说快不快吧。

怎么能让 GPT-3
稍微「动动脑子」呢？想想我们上学的时候老师是怎么做的。

一般来说，优秀的老师会在我们做错题时鼓励我们「再回去想想」，或者帮我们理清解题步骤。同样的，我们也可以这么对 GPT-3。

东京大学和谷歌大脑的一项联合研究表明，只要在答案前加一句「Let’s think step by step」，GPT-3
就能给出上述问题的正确答案，而且它还自己理清了解题步骤。

在经典的 MutiArith 数据集上，这句魔法一样的「咒语」将 GPT-3
在零样本场景下解数学题的准确性从 17.7% 提升到了 78.7%。

重要的是，这句「咒语」的应用范围还非常广泛，不仅可以解数学题，还能做各种逻辑推理。

当然，在深度学习领域，这种「咒语」有个统一的名字——「prompt」。

prompt 和 prompt
工程是近年来非常火的话题，它可以帮助我们控制模型的预测输出。合适的 prompt 对于模型的效果至关重要，大量研究表明，prompt 的微小差别，可能会造成效果的巨大差异[1]。比如在下面这个卫星图片识别的例子中，我们仅添加一个「 satellite（卫星）」，就能把模型的准确率提升 13%+。

图源：《 Learning to Prompt for Vision-Language Models 》

不过这一次，东京大学和谷歌大脑的研究者又把 prompt
的妙用推到了新的高度。

论文链接：https://arxiv.org/pdf/2205.11916.pdf

佐治亚理工学院 ML 博士 Aran Komatsuzaki
开玩笑说，这说明了「『Let’s think step by step』is all you need」。

同时，这也提醒我们，大模型的能力似乎还没有被完全挖掘出来。

不过，对于「Let’s think step by
step」为何如此奏效，谷歌大脑研究科学家 Denny Zhou
指出，这些参与测试的 GPT-3 模型（Text-davinci-002 (175B) 或其他 002

模型或 instruct GPT）可能已经用「let's think step by step. .....」进行过微调。

对此，来自谷歌的论文作者 Shane Gu 表示，instruct GPT 部分确实帮助很大，但我们也能在原始 GPT 上看到很大的提升。

以下是论文细节。

研究概览

大型预训练语言模型被广泛应用于 NLP 的很多子领域。它们具有优秀的少样本甚至零样本学习能力，可以借助几个示例（少样本）或描述任务的一些说明（零样本）简单地给模型施加条件来适应很多任务。这种调节语言模型的方法被称为「prompting」，手工 / 自动设计 prompt 已经成为 NLP 领域的热门话题。

与 LLM 在直观的单步骤 system-1 任务（带有特定于任务的少样本或零样本 prompting）中的出色表现相比，即使是 100B 或更多参数的语言模型在需要缓慢和多步骤推理的 system-2 任务中也表现不佳。（system-1 和 system-2 是心理学家 Keith Stanovich 和 Richard West 率先提出的两种认知过程，前者对应快思考，是由直觉引导的，无意识且快速，比如看到一个数学题一下就反应出是个乘法式；后者对应慢思考，用于解决具有复杂逻辑性的问题，比如解数学题。）

为了解决大模型在 system-2 任务中表现不佳的问题，Wei et al. [2022]、Wang et al. [2022]提出了 chain of thought prompting (CoT)，它为 LLM 提供了 step-by-step 的推理示例，而不是标准问答示例，区别如下图所示。

图源：《 Chain of Thought Prompting Elicits Reasoning in Large Language Models》

CoT 的思维链演示有助于模型生成一个推理路径，该路径将复杂的推理分解为

多个简单的步骤。值得注意的是，有了 CoT 的加持，模型的推理性能更加满足 scaling law，并随着语言模型的规模增长而增长。例如，当与 540B 参数的 PaLM 模型相结合时，与标准少样本 prompting 相比，chain of thought prompting 在多个基准推理任务上显著提升了性能（如在 GSM8K 上从 17.9% 提升到 58.1%）。

虽然 CoT prompting 的成功和许多其他特定于任务的 prompting 工作经常被归功于 LLM 的少样本学习能力，但本文作者表明，通过添加一个简单的 prompt，即「Let’s think step by step」，LLM 就能成为一个优秀的零样本推理器，它会引导模型在给出答案之前展开一步一步的思考（如图 1 所示）。

虽然看起来非常简单，但作者提出的 Zero-shot-CoT 成功地以零样本的方式生成了可行的推理路径，而且最后得到了正确答案，而标准的零样本方法（图 1c）并没有给出正确答案。重要的是，这个 Zero-shot-CoT 非常通用，且不针对具体任务，这不同于之前大多数以示例（少样本）或模板（零样本）的形式进行、特定于某个任务的 prompt 工程。它可以在包括算术、符号推理、常识推理、策略 QA 在内的各种逻辑推理任务中促使模型逐步回答问题，无需为每个任务专门修改 prompt。

如图 1 所示，研究者将 Zero-shot-CoT 与其他 prompting 基线进行了比较。虽然 Zero-shot-CoT 的表现不如有着精心设计的、针对特定任务的 step-by-step 示例的 Few-shot-CoT，但与零样本基线相比，Zero-shot-CoT 实现了巨大的分数提升（在 MultiArith 上从 17.7% 提升到 78.7% ，在 GSM8K 上从 10.4% 提升到 40.7%）。重要的是，与少样本 CoT 基线相比，使用研究者设计的单个固定 prompt，零样本 LLM 会拥有更优秀的 scaling 曲线。

此外，研究者还发现，Few-shot-CoT 除了需要多步骤推理 prompt 的手工工程之外，当 prompt 示例问题类型和任务问题类型不匹配时，它们的表现会下降，这表明它们对逐任务 prompt 设计的敏感性很高。相比之下，研究者提出的单个 prompt 通用性很强，适用于多种推理任务，这表明 LLM 的零样本基础能力还没有被

完全开发出来，比如更高层次的广泛认知能力（如通用逻辑推理）。

研究细节

Zero-shot-CoT 是一个基于零样本模板的 prompting 方法，用于思维链推理。它不同于最初的思维链 prompting [Wei et al., 2022]，因为它不需要 step-by-step 少样本示例，它也不同于之前的大多数模板 prompting，因为它本质上与任务无关，可以通过单一模板在广泛的任务范围内进行 multi-hop 推理。该方法的核心思想非常简单，如上图 1 所示：添加「Let’s think step by step」，或者类似的文本（如下表 5 所示），然后就能让模型进行一步一步的推理。

两阶段 prompting

Zero-shot-CoT 在概念上很简单，其微妙之处在于它使用了两次 prompting，如图 2 所示。这是因为零样本基线（图 1 左下角）已经以「The answer is」的形式使用了 prompting，以正确的格式提取答案。少样本 prompting（standard 或 CoT）通过显式地设计以这种格式结尾（见图 1 右上角）的少样本示例答案来避免对此类答案提取 prompting 的需要。总而言之，Few-shot-CoT [Wei et al., 2022] 需要仔细地人为设计一些 prompt 示例，每个任务都有特定的答案格式，而 Zero-shot-CoT 不需要这样的工程，但需要两次 promp。

第一个 prompt：推理提取。在这一步中，首先使用一个简单的模板「Q: [X]. A: [Z]」将输入问题 x 修改为一个 prompt，其中，[X]是 x 的一个输入位置，[T] 是手工触发的句子 t 的位置，它将提取一个思维链来回答问题 X。例如，如果我们使用「Let’s think step by step」作为触发句，prompt 就是「Q: [X]. A: Let’s think step by step.」然后将被加工成 prompt 的文本 x’ 输入到语言模型中，生成后续句子 z。此处可以使用任何解码策略，但为了简单起见，研究者在整个论文中都使用了贪婪解码。

第二个 prompt：答案提取。在第二步中，使用生成的句子 z 和被加工成

prompt 的句子 x’ 从语言模型中提取最终答案。具体来说，我们简单地将三个元素连接起来，如 [X’] [Z] [A]：[X’] 表示第一个 prompt x’ ，[Z]表示第一步生成的句子，[A]表示用来提取答案的触发句。这一步的 prompt 是自增强的，因为 prompt 包含同一个语言模型生成的句子 z。在实验中，研究者会根据答案格式的不同使用不同的答案触发句。例如，他们在多项选择 QA 中使用「Therefore, among A through E, the answer is」，在需要数字答案的数学问题中使用「Therefore, the answer (arabic numerals) is」。最后，将被加工成 prompt 的文本作为输入馈入语言模型，生成句子?y 并解析最终答案。

实验结果

该研究在四类推理任务的 12 个数据集上评估了新方法，包括算术、常识、符号和其他逻辑推理任务。

该研究对下表所示的 13 种模型进行了实验：

Zero-shot-CoT vs Zero-shot

下表 1 比较了每个数据集上新方法 (Zero-shot-CoT) 和标准零样本 prompting 方法（Zero-shot）的准确性。Zero-shot-CoT 在四种算术推理任务（MultiArith、GSM8K、AQUA、SVAMP）、所有符号推理任务和所有逻辑推理任务上都显著优于 Zero-shot 方法。

该研究还将 Zero-shot-CoT 方法与其他基线进行了比较，在两个算术推理基准（MultiArith 和 GSM8K）上的结果如下表 2 所示。标准 prompting（第一部分）和 thought prompting（第二部分）之间的巨大差距表明，如果不使用多步骤推理，这些任务是非常困难的。

Zero-shot-CoT 自然不如 Few-shot-CoT，但它甚至在每个任务 8 个样本的情况下都能大大优于标准的 few-shot prompting 方法 。对于 GSM8K，使用 Instruct GPT-3 (175B) 的 Zero-shot-CoT 也优于微调

GPT-3 和使用大型模型 (PaLM, 540B) 的标准 few-shot prompting 方法（上表 2 第三部分）。

然后，该研究进一步实验来回答如下几个问题。

模型大小对于零样本推理是否重要？
为了回答这个问题，该研究比较了各种语言模型在 MultiArith 数据集上的性能，结果如下表 3 所示。

如果没有思维链推理（chain of thought reasoning），性能不会随着模型规模的增加而增加，或者只是缓慢地增加，增长曲线大多是平坦的。相比之下，随着模型规模变大，性能随着思维链推理而迅速提升。当模型规模较小时，思维链推理无效。这一结果与 Wei et al. [2022]的 few-shot 实验结果一致。此外，研究者还手动查看了生成的思维链的质量，大模型有着更好的推理效果。

prompt 的选择对 Zero-shot-CoT 方法有什么影响？该研究针对输入 prompt 验证了 Zero-shot-CoT 的稳健性。表 5 总结了使用多个不同句子模板的性能。结果表明，如果以「鼓励」思维链推理的方式编写文本，性能就会得到提升。但是，根据句子的不同，准确性的差异很大。在这个实验中，「Let’s think step by step」达到最佳效果。有趣的是，研究者发现不同模板鼓励模型推理的方式截然不同。

prompt 的选择对 Few-shot-CoT 有什么影响呢？表 6 显示了 Few-shot-CoT 在使用来自不同数据集的样本时的性能。令人惊讶的是，来自不同域但具有相同答案格式的思维链样本提供了相对于 Zero-shot 的显著性能提升。相比之下，当使用具有不同答案类型的样本时，性能增益变少，这表明 LLM 主要利用 few-shot
样本来推断重复格式，而不是任务语境。尽管如此，这两种情况的结果都比 Zero-shot-CoT 差，这明特定任务样本工程对 Few-shot-CoT 是至关重要的。

参考链接：

[1] https://zhuanlan.zhihu.com/p/399295895?utm_source=wechat_session&utm_medium=social&utm_oi=565603530178 56&utm_campaign=shareopn