大语言模型到底能做什么?

刨根问底：大语言模型的训练目标

大语言模型的训练目标是让模型尽可能准确地预测下一个词，从而生成生成自然、连贯的文本。同时，**大语言模型的训练目标实质上也是在构建一个高度压缩的世界知识库。**通过海量文本数据的训练，模型将语言符号与现实世界中的概念、关系、事件等紧密联系起来，形成一个庞大的语义网络。这种压缩能力使得模型能够理解和生成复杂的文本，最后产生超出原有语料的推理和创造，即涌现能力。

引爆技术革命的真实原因：大语言模型的涌现能力

什么是涌现能力（Emergent Capabilities）

根据许多文章的标题，我们会觉得模型的能力是随着参数规模的增加而线性提升的。但事实上，**涌现能力是大语言模型在足够大的规模和质量足够好的数据下，自发产生的一种超出预期的能力。**这种能力并非来自个体的简单叠加，而是系统整体复杂性涌现的结果。这就像水分子在特定温度下突然凝结成冰一样，是一种非线性的质变。涌现能力也揭示了模型的技术潜力，这也是大语言模型火爆的根本原因。

涌现能力的具体表现

- **对话能力**：尽管对话不是模型的原生能力，但通过涌现能力展现了出色的对话能力。
- **上下文学习能力**：无需额外训练，仅通过少量示例即可快速适应新的任务。
- **指令遵循能力**：理解并执行人类的复杂指令，完成特定任务。
- **逻辑推理能力**：进行简单的逻辑推理，解决一些需要思考的问题。
- **知识运用能力**：利用已有的知识库，回答各种问题，甚至进行创造性的内容生成。

大语言模型的涌现能力并非来自预先设定的规则，而是来自模型内部自发形成的复杂结构和关联。类似于生物大脑的进化过程，在足够复杂的神经网络中，新的功能和特性会自然涌现。

大语言模型的能力分类

1. **原生能力**：
    - 文本创造：稿件、邮件、小说、新闻、诗歌等。
2. **涌现能力**：
    - 对话、编程、翻译、逻辑推理（包括自然语言推理等）、文本分类、情感识别。
    - **知识提取与整合能力**：从海量文本中提取信息，并将其整合为可用的知识。
    - **知识运用与推理能力**：利用已有的知识，进行推理、判断、决策，甚至创造新的知识。

如何应用和激发大语言模型的能力?

三种关键方法与应用

1. **提示工程（Prompt Engineering）**
    - 通过精心设计的提示语，引导模型按照我们的意图生成内容或完成任务。
2. **微调（Fine Tuning）**
    - 在预训练模型的基础上，使用特定领域的数据进行训练，使模型在特定任务上表现更好。

3. **构建智能机器人代理（agent）**
    - 以大语言模型为大脑驱动的系统，具备自主理解、感知、规划、记忆和使用工具的能力，能够自动化执行完成复杂任务的系统。
        - **听起来很复杂，实际上就是将大模型的智能能力在不止是对话聊天机器人的场景释放出来，或者说对话聊天机器人就是个agent，我让大模型操控电脑就可以是个agent（已经实现了 如Claude的computer use）那么这样一整个的agent项目中，可能涉及多次的和大模型对话的部分，这可以看成是每一个子agent，于是每个子agent就有了自己的侧重点和上下游的分工，比如这个agent分解任务，那个agent产生具体想法，另外的agent反思等等**

因为微调改变了模型本身的参数，这听起来可能更高深一点，让人认为微调比提示工程更强大。但事实上，**提示工程具有更高的优先级和灵活性可解释性。**
如果可以用prompt解决，尽量用prompt解决，因为训练（精调）的模型往往通用能力会下降，训练和长期部署成本都比较高，这个成本也包括时间成本。并且**如果写prompt就可以达到基本要求，那么微调可以进一步提升；如果prompt不起作用，微调成功的可能性就很低。**
同时，提示工程更是**一种引导模型解决复杂问题的方法论。**这就像构造agent时的架构与指挥官。

提示工程可以看作是一种"软微调"和Agent的构造框架，通过改变Prompt，间接影响模型的行为，**提示工程不仅仅是技术，更是一种思维方式。** 它要求我们从模型的角度思考问题，理解模型的认知模式，才能有效地引导模型。

提示工程技术：深入理解与学习路径

提示工程的误解

- **误区**：
    - 简单添加提示词，如"请一步步解答"或"参照示例回答"。提示词模板化，变通性少技术含量低。
- **真实场景的提示工程**：
    - **核心能力**：**一种思维方式。** 它要求我们从模型的角度思考问题，理解模型的认知模式，才能有效地引导模型。
    - **价值**：结合人工经验与技术灵感解决复杂问题，是前沿研究热点。

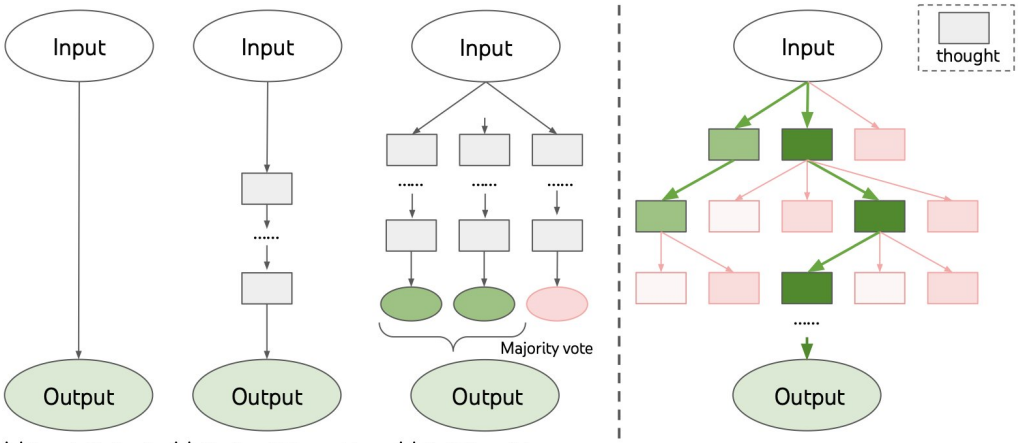
我们可能会认为提示工程是一种临时的解决方案，随着模型能力的提升，提示工程将不再重要。但从历史发展到如今的o1来看，**提示工程及其提倡的理念将长期存在，并朝着更加自动化、智能化的方向发展。**

提示工程两大目的方向

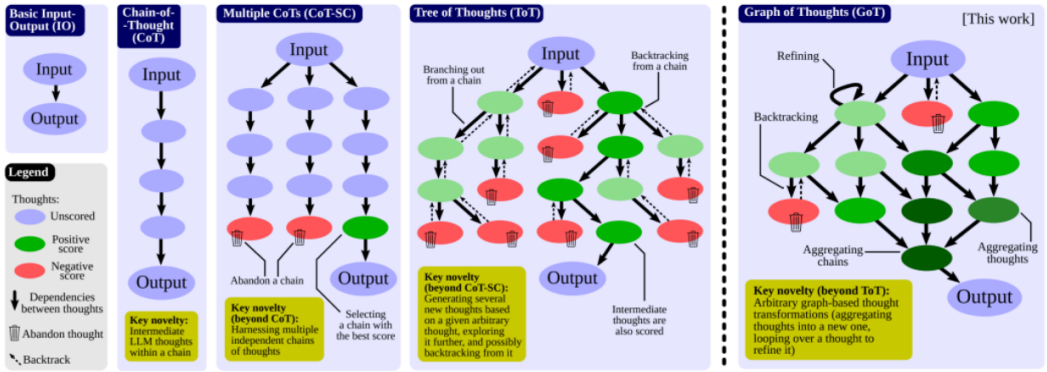
1. 限定模型的回复为特定格式，让大模型听话
    - Json结构等自己想要的生成结构
2. 发挥大模型的推理及复杂任务处理能力，Agent可以理解为thought+action，提示工程就是在规划thought
    - COT（chain of thougnt）,TOT,GOT等

提示工程三大核心技术

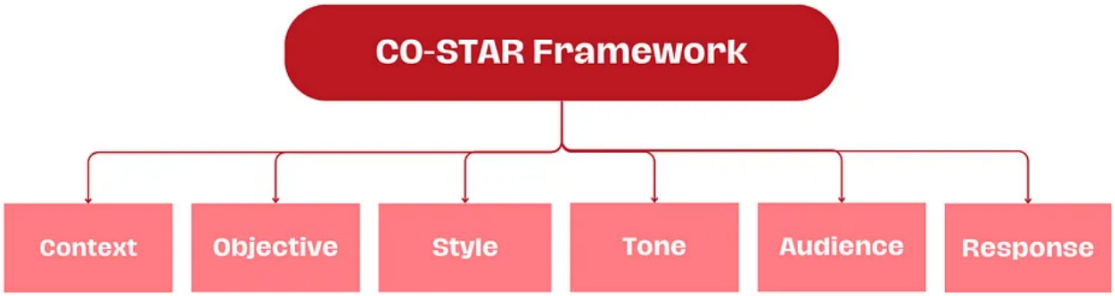
1. **提示词模板设计**：
   - 设计“魔法语句”，如“请逐步思考”。
2. **提示示例设计**（few shot）：
   - 包括问题、答案及推理步骤，引导模型更好地完成任务。
3. **提示流程设计**（COT，TOT，GOT）：
   - 设计多步提示流程，逐层解决复杂问题。
     TOT



GOT



4. **提示词模板设计**：
   为了让 LLM 给出最优响应，为 prompt 设置有效的结构至关重要。CO-STAR 框架是一种可以方便用于设计 prompt 结构的模板，这是新加坡政府科技局的数据科学与 AI 团队的创意成果。该模板考虑了会影响 LLM 响应的有效性和相关性的方方面面，从而有助于得到更优的响应。



CO-STAR 框架

- (C) 上下文（Context）推荐：提供与任务有关的背景信息。这有助于 LLM 理解正在讨论的具体场景，从而确保其响应是相关的。
- (O) 目标（Objective）推荐：定义你希望 LLM 执行的任务。明晰目标有助于 LLM 将自己响应重点放在完成具体任务上。
- (S) 风格（Style）可选：指定你希望 LLM 使用的写作风格。这可能是一位具体名人的写作风格，也可以是某种职业专家（比如商业分析师或 CEO）的风格。这能引导 LLM 使用符合你需求的方式和词语给出响应。
- (T) 语气（Tone）可选：设定响应的态度。这能确保 LLM 的响应符合所需的情感或情绪上下文，比如正式、幽默、善解人意等。
- (A) 受众（Audience）可选：确定响应的目标受众。针对具体受众（比如领域专家、初学者、孩童）定制 LLM 的响应，确保其在你所需的上下文中是适当的和可被理解的。
- (R) 响应（Response）可选：提供响应的格式。这能确保 LLM 输出你的下游任务所需的格式，比如列表、JSON、专业报告等。对于大多数通过程序化方法将 LLM 响应用于下游任务的 LLM 应用而言，理想的输出格式是 JSON。

同时，特定的风格角色和任务也可以使用提示词改进器或让大模型帮忙改进

OpenAI通过提示工程策略增强大型语言模型（如GPT-4）输出结果的指南文章里提出了六种主要策略：

1. **清晰的指示**：模型无法读懂你的想法，因此需要清晰具体的指令，包括具体的目标、细节、期望的格式等。
2. **提供参考文本**（few shot）：提供参考文本可以帮助模型更准确地回答问题，避免捏造信息。
3. **将复杂任务分解成更简单的子任务**（COT、TOT、GOT、agent）：将复杂任务分解成一系列简单的步骤，可以提高模型的准确率并降低错误率。
4. **给模型时间"思考"**：鼓励模型进行推理，避免匆忙得出结论，例如，可以要求模型先给出自己的解决方案，再与提供的方案进行比较。
5. **使用外部工具**（tool use）：利用外部工具弥补模型的不足，例如使用嵌入式搜索进行知识检索，或使用代码执行引擎进行计算。
6. **系统地测试更改**：通过评估模型输出来衡量提示修改的效果，确保改进是积极的。

https://platform.openai.com/docs/guides/prompt-engineering#six-strategies-for-getting-better-results