# 【LLM学习】Applied LLMs：LLMs构建应用程序的实践经验总结（3）战略层面

方方 方方的算法花园 2024年06月16日 23:11 浙江

“本文主要针对《What We’ve Learned From A Year of Building with LLMs》文章进行了翻译和总结，原文是一个非常实用的指南，介绍如何利用LLMs构建成功的产品，文章内容比较长，我会分成战术应用、运营、战略三篇文章进行解读。”

原文地址：https://applied-llms.org/（发布时间：June 8, 2024）

作者：Eugene Yan、Bryan Bischof、Charles Frye、Hamel Husain、Jason Liu、Shreya Shankar

【LLM学习】Applied LLMs：LLMs构建应用程序的实践经验总结（1）战术应用

【LLM学习】Applied LLMs：LLMs构建应用程序的实践经验总结（2）运营策略

成功的产品建立在深思熟虑的规划和明智的优先级排序之上，而不是无休止的原型迭代或盲目追逐最新技术趋势。在本节中，我们将深入分析构建卓越人工智能产品的关键战略要素。同时，我们将评估团队在开发过程中可能面临的主要权衡，例如决定是自主研发还是外部采购。此外，我们将提出一套针对早期大型语言模型应用开发的策略指南，旨在指导团队高效地实现产品目标。

## 01 产品市场契合前不急于使用 GPU

卓越的产品开发不仅仅是简单地依赖外部API进行表层封装。错误的策略选择往往伴随着高昂的代价。回顾过去一年，风险投资的热潮催生了巨额资金的注入，例如高达六十亿美元的A轮融资。然而，这些资金往往被用于模型训练和定制，却缺乏明确的产品定位和目标市场策略。在本节中，我们将深入讨论为何急于自主训练模型可能并非明智之举，并探讨自我托管模型的重要性及其潜在优势。

### 1.1 几乎没有必要从头开始训练

对于绝大多数组织而言，从头开始预训练一个大型语言模型（LLM）往往是一种资源的浪费。尽管这一做法看似具有吸引力，似乎每个组织都在尝试，但开发和维护机器学习基础设施实际上非常耗费资源。这包括数据收集、模型训练与评估，以及最终的部署过程。如果产品还处于市场验证阶段，这些工作可能会分散你的注意力，使你无法专注于核心产品的研发。

以金融领域的BloombergGPT为例，该模型由九位全职员工（四名AI工程师和五名机器学习产品及研究人员）共同开发，基于3630亿个数据令牌进行预训练。然而，不到一年的时间，它就被新一代的模型如gpt-3.5-turbo和gpt-4在相同任务上超越。

这一例子以及其他类似情况表明，对于大多数实际应用，从头开始预训练大语言模型，哪怕是针对特定领域的数据，也并非最佳选择。团队应该更倾向于对现有的强大开源模型进行微调，以满足自身的特定需求。

然而，也有一些例外情况。例如，Replit的代码模型，这是一个专为代码生成和理解而设计的模型。通过预训练，Replit在某些方面成功超越了其他大型模型，如CodeLlama7b。但是，随着越来越多高效的模型问世，维持其有效性需要不断的投入和更新。

### 1.2 微调：非必要，不急于行动

在许多情况下，组织对微调的热衷往往源于对错失机会的担忧，而非基于深思熟虑的战略规划。避免过早微调的诱惑至关重要。微调应被视为一种重要的工具，仅在其他方法明显不足且有充分数据支持的情况下才应考虑使用。

回顾过去一年，许多团队在微调上的尝试并未带来预期的产品市场契合度，反而留下了遗憾。如果决定进行微调，必须准备好面对基础模型持续优化所带来的挑战，这需要参考后续章节中关于"模型不是产品"和"构建LLMOps"的讨论。

何时微调才是合适的选择？当特定应用场景所需的数据无法从现有模型训练所用的广泛开放网络数据集中获得，并且你已经通过最小可行产品展示了现有模型的不足时。然而，必须谨慎行事：如果连模型开发者都难以获取高质量的训练数据，那么获取这些数据的难度可想而知。

由大型语言模型支持的应用不应仅仅停留在科学实验阶段，它们的投资应与对企业战略目标和市场竞争力的实际贡献相匹配。

### 1.3 推理API与自建服务：权衡与选择

利用大语言模型的推理API，创业公司可以快速集成语言模型功能，无需承担从头训练模型的重负。公司如Anthropic和OpenAI提供的通用API使产品智能化变得简单，仅需几行代码。这种方法允许你节省宝贵的资源，专注于为客户创造价值，快速验证创意，并实现市场适配。

然而，随着业务规模的增长和特定需求的增加，依赖外部托管服务可能不再是最优解。在医疗、金融等受严格监管的行业，或在需要遵守合同和保密要求的情况下，自建服务成为确保敏感数据安全的唯一选择，确保数据不离开内部网络。

自建服务还提供了避免外部推理服务提供商限制的优势，如使用频率限制、模型更新的不确定性和使用约束。此外，自建服务使你能够完全控制模型，更容易定制和优化，以构建具有特色的高质量系统。在大规模部署时，自建服务还能显著降低成本。Buzzfeed通过微调开源的大语言模型，成功减少了80%的成本，这一案例证明了自建服务在成本效益上的潜力。

## 02 不断迭代，迈向卓越

为了长期保持竞争力，你需要考虑的不仅是模型本身，还要思考如何使你的产品独一无二。执行速度虽重要，但不应是你唯一的优势。

## 2.1 模型与产品：超越模型本身

在技术创新的浪潮中，不直接构建模型的团队可以利用快速进步的模型技术，不断追求在上下文处理、逻辑推理和性价比上的提升，使产品更加完善。这种进步不仅令人兴奋，而且是可以预见的。在产品系统中，模型往往是最容易更新和迭代的部分。

然而，真正的产品价值来自于那些能够带来长期价值的系统组件，包括：

- **性能评估（Evals）**：确保跨不同模型的一致性和可靠性。
- **安全防护（Guardrails）**：确保模型输出符合预期，避免不希望的结果。
- **数据缓存（Caching）**：通过减少对模型的直接调用，降低延迟和成本。
- **数据驱动（Data flywheel）**：通过持续的数据反馈循环，推动所有环节的改进。

这些组件构成了产品质量的坚实防线，比单一的模型更加持久和可靠。

但这并不意味着在应用层面的开发没有风险。团队应避免在与模型供应商如OpenAI等重复劳动的方向上浪费精力，因为这些供应商通常会处理这些通用需求。

例如，一些团队可能会投资开发自定义工具来校验模型输出的结构化数据。适度的投资是必要的，但过度投资可能会导致时间和资源的浪费。OpenAI和其他供应商会确保基本的函数调用正确无误，因为这是所有客户的共同需求。在这种情况下，团队可以采取"战略性推迟"的策略，只构建必要的部分，并等待供应商技术的明显进步。

## 2.2 构建信任：专注的力量

追求开发一个万能的产品往往会导致平庸无特色。要创造真正吸引人的产品，公司需要专注于创造一种吸引力体验，使用户愿意不断回访。

以一个旨在回答所有问题的通用RAG（Retrieval-Augmented Generation）系统为例。这种系统由于缺乏专注，无法优先处理最新信息、解析特定领域的格式或充分理解特定任务的复杂性。这导致用户得到的是一个表面且不稳定的体验，难以满足他们的需求，最终可能导致用户流失。

为了解决这个问题，公司应该专注于特定领域和具体的应用场景，深耕细作而不是广撒网。这种方法不仅可以开发出与用户需求高度契合的领域工具，而且可以清晰地向用户传达系统的功能和局限。通过提高系统的透明度，公司不仅能增强自我认知，还能让用户明确知道系统在哪些方面能为他们带来最大的价值。这种透明度有助于建立用户的信任和信心，是长期成功的基石。

## 2.3 构建大语言模型运维 (LLMOps)：追求快速迭代

DevOps的核心不仅仅在于建立可复制的工作流程或赋能小团队，它远比编写YAML文件复杂得多。DevOps的真正价值在于缩短工作与结果之间的反馈周期，实现持续的优化积累，而不是错误积累。这一理念源自精益生产和丰田生产系统，强调快速换模和持续改进（Kaizen）。

MLOps将DevOps的理念应用于机器学习，虽然提供了可复现的实验和一体化工具套件来支持模型开发，但它并没有有效缩小模型在实际应用中的反馈延迟。

然而，大语言模型运维（LLMOps）正在从处理如提示管理这样的小问题转向更严峻的挑战：通过生产监控和持续改进来实现评估连接。LLMOps的目标是实现更快的迭代，以适应快

速变化的技术和市场需求。

目前，市场上已经出现了如LangSmith、Log10、LangFuse、W&B Weave、HoneyHive等工具，它们不仅能收集和整理生产数据，还能与开发过程深度融合，利用这些数据来优化系统。组织可以选择使用这些现成的工具，或者根据特定需求自行开发解决方案。

## 2.4 避免自行开发可购买的LLM功能

虽然大多数成功的企业并不直接依赖大语言模型（LLM）来运营，但LLM无疑能显著提升业务效率。然而，这种认识常常误导领导层急于将LLM整合进现有系统，导致成本增加和效率降低，同时推出那些已经被市场过度饱和的“闪光AI”功能。

**理智的方法是：** 专注于那些真正符合你的产品目标并能够显著提升核心业务的LLM应用。

以下是一些常见的陷阱，它们浪费了团队的时间和资源：

- 开发定制的文本到SQL（text-to-SQL）功能，而市场上已有成熟解决方案。
- 创建与公司文档对话的聊天机器人，却忽视了现有工具的潜力。
- 将公司知识库与客户支持聊天机器人整合，而没有评估现成解决方案的可行性。

尽管这些是LLM应用的基础示例，但对于产品公司来说，自行开发它们并不明智。从有前景的演示到可靠的产品组件之间，存在着巨大的鸿沟，这通常是软件公司的主战场。将宝贵的研发资源投入到这些问题中，可能会分散对核心业务的关注。

如果这听起来像是陈词滥调的商业建议，那是因为在当前的炒作浪潮中，很容易被“大语言模型（LLM）”这样的前沿标签所迷惑，而忽视了那些已经成为常态的应用。在LLM的热潮中，保持清醒的头脑，专注于那些真正能够为公司带来长期价值的解决方案，是至关重要的。

## 2.5 人工智能辅助，人类主导

目前，由大语言模型驱动的应用虽然功能强大，但仍然存在脆弱性。它们需要严格的安全防护和精心设计的防御机制，尽管如此，预测其行为的难度依然很大。然而，当这些应用的功能定位精准时，它们可以显著提高用户的工作流程效率，证明大语言模型是加速任务执行的优秀工具。

尽管可以想象一个完全由大语言模型驱动的应用取代某个工作流程或职能，但在当前技术阶段，最有效的模式是人机协作模式，类似于"半人马象棋"。当能力出众的人类与专为快速应用而优化的大语言模型功能结合时，可以显著提升执行任务的效率和满意度。GitHub CoPilot的成功案例已经证明了这一点：

"开发者们反馈，使用GitHub CoPilot后，他们在编程时感觉更自信，因为编码过程更简单、更少出错、更易于阅读、更易于重用、更加简洁、更易维护，且更具有弹性。" - Mario Rodriguez, GitHub

对于长期从事机器学习的专家来说，"人在循环中"的概念可能并不陌生，但我们在这里强调的是更为细微的差别。大语言模型驱动的系统不应成为现今大部分工作流程的主导力量，而应作为一种辅助资源，支持和增强人类的能力。

将人类置于核心，探询大语言模型如何辅助他们的工作流程，可以引发截然不同的产品和设计决策。这种方法最终会促使你开发出与那些急于将所有责任推给大语言模型的竞争对手不同的产品——更优质、更实用、风险更低的产品。

# 03 提示、评估和数据收集

如果一个团队想构建一个大语言模型 (LLM) 产品，他们应从何入手？过去一年的实践已经足以证明，成功的大语言模型应用都遵循着一条一致的轨迹。本节将介绍这套基本的入门策略。其核心思想是从简单做起，仅在必要时增加复杂性。一个实用的规则是，每增加一层复杂度，通常需要的努力至少是前一层的十倍。

## 3.1 提示工程：优化AI交互的第一步

在开发由大语言模型驱动的应用时，**提示工程**应成为你的首要任务。利用我们在策略部分讨论的技术，如思维导图、少样本示例以及结构化的输入与输出，可以有效地引导模型理解并执行所需的任务。这些方法不仅能够提高模型的性能，还能确保它按照预期的方式工作。

在原型设计阶段，应优先使用**最先进的模型**来验证你的概念。这不仅可以帮助你快速迭代，还能确保你的应用能够利用当前最佳的技术成果。

**仅在提示工程无法达到预期性能水平时**，才应考虑进行模型的微调。微调是一个资源密集型的过程，应该作为最后的手段，仅在其他方法无法满足需求时采用。

如果存在非功能性需求，如数据隐私、完全控制权或成本问题，这可能会促使你选择自行托管模型。在这种情况下，微调可能会更常见，但必须确保这些隐私需求不会妨碍你使用用户数据来进行微调。找到平衡点，确保既能满足隐私要求，又能利用数据来提升模型性能。

通过将提示工程作为起点，你可以在不牺牲性能的前提下，最大限度地减少对资源的依赖，同时确保应用的灵活性和可扩展性。

## 3.2 构建评估体系，启动数据驱动飞轮

即使是初创团队，也必须将评估工作纳入产品开发的核心流程。没有评估，就无法准确判断提示工程的效果，或确定微调模型何时能够取代基础模型。

**有效的评估策略** 应该专注于具体任务，并能够真实反映产品在实际使用中的表现。我们建议从**单元测试**开始，这种基础测试有助于在早期阶段发现问题并指导设计决策。此外，还可以探索针对特定任务的评估方法，如文本分类、摘要生成等。

虽然单元测试和基于模型的评估至关重要，但它们不能替代**人工评估**的重要性。让用户直接体验模型或产品，并收集他们的反馈，不仅可以评估产品在现实世界中的表现，发现潜在缺陷，还可以收集宝贵的数据，为未来的模型微调提供支持。这种人工评估形成的正向反馈循环，即数据驱动飞轮，将随着时间的积累而不断增强：

1. 通过人工评估测试模型性能或识别问题。
2. 利用收集到的数据进行模型微调或更新提示。
3. 重复这一过程，不断优化。

例如，在检查由大语言模型生成的摘要时，可以对每个句子进行详细反馈，指出事实错误、不相关内容或风格问题。这些反馈可以用来训练识别错误信息的分类器或评估内容相关性的奖励模型。LinkedIn分享了他们如何使用基于模型的评估器来检测生成内容的准确性、伦理合规性和连贯性。

通过这种方式，评估工作从一项运营成本转变为一种战略投资，帮助我们在产品开发过程中逐步构建并加速数据驱动飞轮的运转。

## 3.3 低成本智能的未来趋势

在1971年，Xerox PARC的先见之明不仅预测了一个由网络化个人电脑构成的世界，而且通过参与Ethernet、图形渲染技术、鼠标和窗口系统等关键技术的发明，为实现这一愿景做出了

实质性贡献。他们的洞察力和创新精神塑造了我们今天所熟知的技术环境。

Xerox PARC的研究者们还采用了一种预测练习，该练习可以为我们今天对大语言模型技术成本下降的预测提供参考。他们首先识别出那些极具价值但成本高昂的技术应用，例如视频显示技术。然后，他们利用这些技术的历史价格趋势，参照摩尔定律，来预测这些技术何时能够变得经济实惠并广泛普及。

对于大语言模型技术，我们可以采取类似的策略来进行未来成本的预测。选择一个广受欢迎的大规模多任务语言理解数据集，配合一致的输入方法（例如，允许模型有五次尝试的机会），然后比较不同性能等级的语言模型在这一基准上的表现和运行成本随时间的变化。

通过这种方法，我们可以更准确地预测大语言模型技术的成本效益比将如何随着时间而改善，以及这些技术何时能够以更低的成本提供更高质量的智能服务。这种前瞻性分析不仅有助于企业制定战略规划，也为技术投资和市场进入提供了宝贵的洞见。

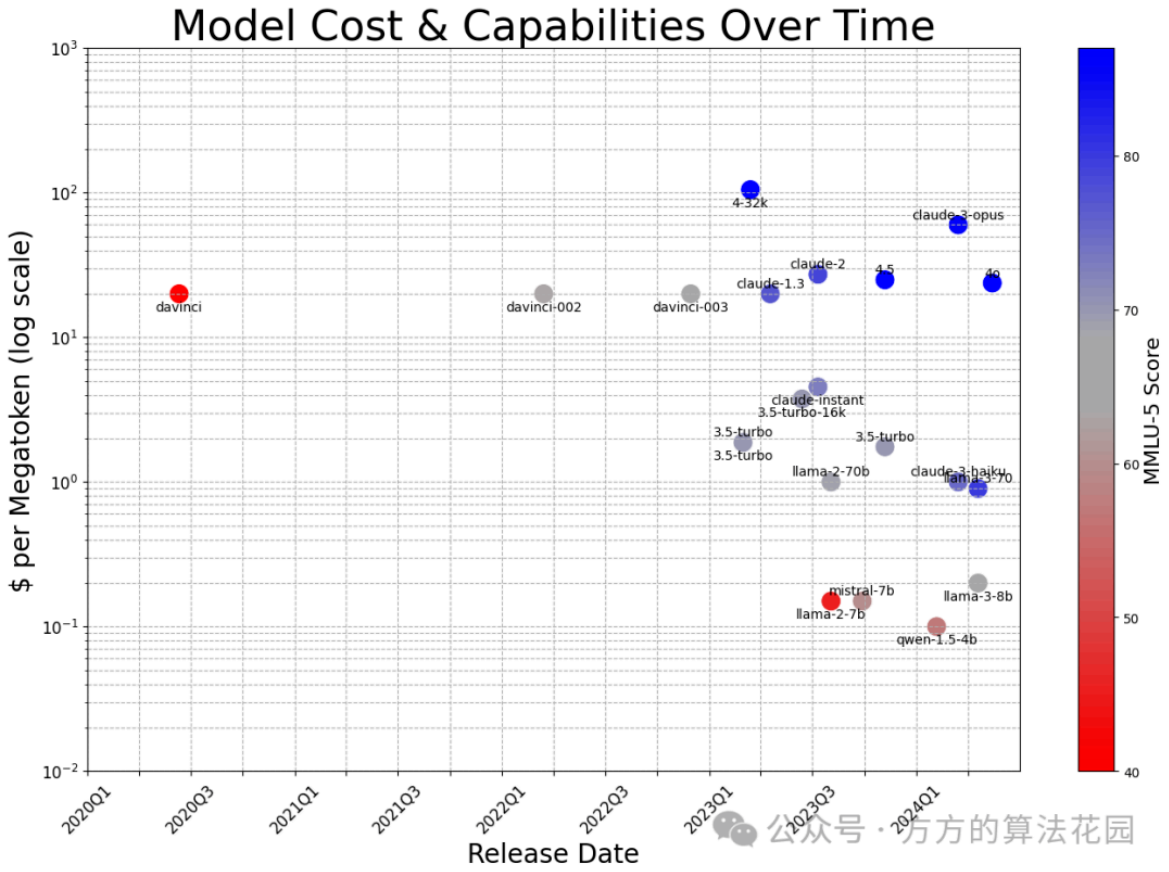


图 1: 固定成本下，模型能力快速提升；在固定能力水平下，成本则在迅速降低。图由合著者 Charles Frye 在 2024 年 5 月 13 日利用公开数据创建。

自OpenAI的davinci模型作为API发布以来，短短四年内，执行相同任务的模型运行成本已经从每百万字符20美元降至不到10美分，减半时间仅为六个月。至2024年5月，Meta的LLaMA 3 8B模型的成本预计仅为每百万字符20美分，与OpenAI的text-davinci-003模型在2023年底发布时的成本相同，但后者的成本是20美元。仅在18个月内，成本已经减少了两个数量级，这一下降速度远超摩尔定律的预测。

**实用应用案例：** 大语言模型（LLM）的一个非常实用的应用是在生成式视频游戏角色的赋能上，类似于Park et al的研究。尽管目前这种应用的成本还不够经济，据估计为每小时625美

元，但自2023年8月该论文发布以来，成本已经大幅下降至每小时62.50美元。预计在未来九个月内，这一成本可能进一步降至每小时6.25美元。

**历史对比：** 回顾1980年推出的吃豆人游戏，当时1美元可以让你玩几分钟到几十分钟的游戏，相当于每小时消费约6美元。通过这一粗略估算，我们可以预见到2025年，引人入胜的大语言模型增强的游戏体验将变得更加经济实惠。

**未来趋势：** 尽管大语言模型的成本下降趋势只出现了几年，但考虑到数据中心和硅层的深层次创新和投资，这一进程在未来几年内减缓的可能性很小。我们可能已经利用了一些算法和数据集中较容易取得的成果，但未来的深层次创新有望补足这一不足。

**战略意义：** 今天看似完全不可行的展示或研究，几年后可能变成高端功能，随后很快普及成为常见产品。这一认识对于构建我们的系统和组织结构至关重要，它要求我们具有前瞻性，为未来的技术变革做好准备。

#LLM学习 12

#LLM学习 · 目录

上一篇
【LLM学习】Applied LLMs：LLMs构建应用程序的实践经验总结（2）运营策略

下一篇
【LLM论文阅读】LlamaRec:具有高效检索与排序的两阶段推荐框架