

高能干货分享，有关提示词工程的一切都在这份教程里

关注LLM的 机器之心 2024年11月10日 11:45 北京

机器之心报道

编辑：佳琪、Panda

想让 AI 帮你自动搞定任务，离不开要在聊天框内用提示词对它发号施令。

但找到合适的提示词并不容易，这需要设计、优化和评估，不仅耗时，还需要自己学习技巧。

有时候，你可能感觉这和调参有着异曲同工之妙，怎么都是玄学？

尝试了不同的措辞、语气，甚至还和模型玩上了角色扮演。但实际上，就像一千个人眼中有一千个哈姆雷特一样，对于完全相同的提示词，模型也会给出不同的回答。因此，我们难以确定哪些方法真正奏效。

今天，开源社区 DiamantAI 的主理人 Nir Diamant 发布了一套提示词工程技术库，系统性地教我们如何提高和 AI 的沟通技巧，更好发挥 AI 的潜能。



才上线几个小时，它就在 GitHub 上拿到了 200 多颗 star，并且还在以这个速度不断攀升。

Product Solutions Resources Open Source Enterprise Pricing

Search Sign in Sign up

NirDiamant / Prompt_Engineering Public

Sponsor Notifications Fork 23 Star 249

Code Issues Pull requests Actions Projects Security Insights

main 1 Branch 0 Tags Go to file Code

NirDiamant updated CONTRIBUTING 0594d88 · 16 hours ago 3 Commits

.github	copy files from private repo	17 hours ago
all_prompt_engineering_techniques	copy files from private repo	17 hours ago
.gitignore	copy files from private repo	17 hours ago
CONTRIBUTING.md	updated CONTRIBUTING	16 hours ago
LICENSE	copy files from private repo	17 hours ago
README.md	updated README	17 hours ago
requirements.txt	copy files from private repo	17 hours ago

About

This repository offers a comprehensive collection of tutorials and implementations for Prompt Engineering techniques, ranging from fundamental concepts to advanced strategies. It serves as an essential resource for mastering the art of effectively communicating with and leveraging large language models in AI applications.

ai tutorials hacktoberfest prompt-engineering langchain genai

Readme View license

无论是像谷歌、微软这样的科技巨头，还是吴恩达等资深专家，以及众多行业大咖都推出过专业的提示词课程。更不用提，通过「魔法打败魔法」，用 AI 自动生成提示词的方法也是一抓一大把，比如这篇文章《还在人工炼丹？自动提示工程指南来了，还带从头实现》。

那么，为什么这个提示词课程还能脱颖而出，收到如此多的关注呢？

答案可能就在作者 Nir Diamant 自己写下的推荐语中：「所有与提示词工程相关的内容，你都能在这里找到！」



NirD

@NirDiamantAI

A FREE goldmine of tutorials about Prompt Engineering!

I've just released a brand-new GitHub repo as part of my Gen AI educative initiative.

You'll find anything prompt-engineering-related in this repository. From simple explanations to the more advanced topics.

🔗 Check it out here: github.com/NirDiamant/Pro...

The content is organized in the following categories:

1. Fundamental Concepts
2. Core Techniques
3. Advanced Strategies
4. Advanced Implementations
5. Optimization and Refinement
6. Specialized Applications
7. Advanced Applications

As of today, there are 22 individual lessons.

The screenshot shows the GitHub repository page for 'Prompt Engineering Techniques: Comprehensive Repository for Development and Implementation' by NirDiamant. The repository is a public repository with 35 stars, 1 watching, and 3 forks. The README file is the most recent, updated 5 hours ago. The repository description states: 'Welcome to one of the most extensive and dynamic collections of Prompt Engineering tutorials and implementations available today. This repository serves as a comprehensive resource for learning, building, and sharing prompt engineering techniques, ranging from basic concepts to advanced strategies for leveraging large language models.' The repository is organized into 7 categories: 1. Fundamental Concepts, 2. Core Techniques, 3. Advanced Strategies, 4. Advanced Implementations, 5. Optimization and Refinement, 6. Specialized Applications, and 7. Advanced Applications. The repository also includes a 'Stay Updated!' section with a link to subscribe to the 'DiamantAI's Newsletter of Gen AI'. The repository is currently in the 'Introduction' section, which describes the goal of the repository: 'Our goal is to provide a valuable resource for everyone - from beginners taking their first steps in AI to seasoned practitioners pushing the boundaries of what's possible. By offering a range of examples from foundational to complex, we aim to facilitate learning, experimentation, and innovation in the rapidly evolving field of prompt engineering. Furthermore, this repository serves as a platform for showcasing innovative prompt engineering techniques. Whether you've developed a novel approach or found an innovative application for existing techniques, we encourage you to share your work with the community.'

10:32 PM · Oct 10, 2024 · 9,138 Views

从教程的体量上，就已经能感受到它有多全面了。它总共分 7 大部分，22 个章节，由易到难逐步进阶，条理分明，从基础概念入门，到最近流行的思维链提示词，鼓励 AI 模型将复杂问题分解为逐步推理过程。每节课都附有详细的代码。

Setup

First, let's import the necessary libraries and set up our environment.

```
In [2]: import os
        from langchain_openai import ChatOpenAI
        from langchain.prompts import PromptTemplate
        from langchain.chains import ConversationChain
        from langchain.memory import ConversationBufferMemory

        from dotenv import load_dotenv
        load_dotenv()

        os.environ["OPENAI_API_KEY"] = os.getenv('OPENAI_API_KEY') # OpenAI API key
        # Initialize the language model
        llm = ChatOpenAI(model="gpt-4o-mini")
```

我们可以看到，教程从最基础的环境设置入手，如 OpenAI API 和 LangChain 库，逐步引导你从零样本提示开始，深入学习如何实现基础和高级的少样本学习，以及如何可重复使用，大大提升效率的结构化提示。

在整个学习的过程中，教程还会教你如何根据模型的响应进行迭代优化，不断提升提示效果。

在具体课程中，Nir Diamant 设计了各种案例，让你直观感受一下，提高自己的沟通技巧，对提升 AI 模型对事实的准确性有什么效果，即使是提示结构和措辞的微小调整，也能帮助 AI 更好地吃透你的需求。

有时，精心设计的提示词可能包含过多条件，导致提示过长，反而增加了模型理解的难度。课程还讲解了如何在提示词的细节和简洁性之间找到平衡。

和其他泛泛讲述提示词工程对等方面有什么应用的课程不同，学到后期，你就可以上手设计跨多语言工作、防止提示词注入并在提示词中实施内容过滤等等高级应用了。

- GitHub 链接: https://github.com/NirDiamant/Prompt_Engineering

课程概览

基础概念

1. **Introduction to Prompt Engineering**，提示词工程入门：在 AI 和语言模型语境中全面介绍提示词工程的基础概念，其中包含理论解释和实际演示，覆盖了基础概念、结构化提示词、比较分析和解决问题的应用。

2. **Basic Prompt Structures**，基础提示词结构：其中探索了两种基本类型的提示词结构，包括单轮提示词和多轮提示词（对话）。其实现使用了 OpenAI 的 GPT 模型和 LangChain，演示了单轮和多轮提示词、提示词模板和对话链。

3. **Prompt Templates and Variables**, 提示词模板和变量: 介绍了如何使用变量来创建和使用提示词模板, 重点是 Python 和 Jinja2 模板引擎。实现上, 其中涉及模板创建、变量插入、基于条件的内容、列表处理、整合 OpenAI API。

核心技术

1. **Zero-Shot Prompting**, 零样本提示: 探索了零样本提示, 让语言模型可以在没有具体示例或先验训练时也能执行任务。具体实现上, 其中演示了直接任务规范设定、基于角色的提示词、格式规范设定和多步骤推理, 使用的工具是 OpenAI 的 GPT 模型和 LangChain。

2. **Few-Shot Learning and In-Context Learning**, 少样本学习和上下文学习: 使用 OpenAI 的 GPT 模型和 LangChain 库介绍少样本学习和上下文学习技术。其实现包括基础和高级的少样本学习、上下文学习, 还有示例选取方法和评估的最佳实践。

3. **Chain of Thought (CoT) Prompting**, 思维链提示: 介绍思维链提示技术, 引导 AI 模型将复杂问题分解为逐步推理过程。其实现涵盖基础和高级思维链技术, 解决各种问题的应用场景, 并还比较了标准提示词技术。

进阶策略

1. **Self-Consistency and Multiple Paths of Reasoning**, 自我一致性和多路径推理: 探索生成多样化推理路径并聚合结果以改进 AI 生成的答案的技术。其实现包括设计多样化推理提示、生成多个响应、实现聚合方法和应用自我一致性检查。

2. **Constrained and Guided Generation**, 有约束和引导的生成: 专注于设置模型输出约束和实现基于规则的生成技术。使用 LangChain 的 PromptTemplate 进行结构化提示, 实现约束条件, 探索基于规则的生成技术。

3. **Role Prompting**, 角色提示: 探索为 AI 模型分配特定角色并制定有效的角色描述。展示如何创建基于角色的提示, 为 AI 模型分配角色, 并针对各种场景优化角色描述。

高级实现

1. **Task Decomposition in Prompts**, 提示词中的任务分解: 探索复杂任务分解技术和提示词中的链式子任务。涵盖问题分析、子任务定义、定向提示词工程、按顺序执行和结果综合。

2. **Prompt Chaining and Sequencing**, 提示链和序列处理: 演示如何连接多个提示词并为复杂的 AI 驱动的任务构建逻辑流程。其实现中探索了基本的提示链、序列提示、动态提示词生成和提示链中的错误处理。

3. **Instruction Engineering**, 指令工程：专注于为语言模型制定清晰有效的指令，平衡专用性和通用性。其实现中涵盖创建和优化指令、实验不同结构，并基于模型响应实施迭代改进。

优化和改进

1. **Prompt Optimization Techniques**, 提示词优化技术：探索提示词优化的高级技术，重点是 A/B 测试和迭代改进。在其实现中，演示了提示词的 A/B 测试、迭代改进过程，以及使用相关指标进行性能评估。

2. **Handling Ambiguity and Improving Clarity**, 处理歧义和提高清晰度：专注于识别和解决模糊提示，以及编写更清晰提示的技术。在其实现中，涵盖分析模糊提示、实现消除歧义的策略，探索了编写更清晰提示的技术。

3. **Prompt Length and Complexity Management**, 提示词长度和复杂度管理：其中探索了使用大型语言模型时管理提示词长度和复杂度的技术。在其实现中，演示了平衡细节和简洁性的技术，以及处理长上下文的策略，包括分块、汇总和迭代处理。

专业化应用

1. **Negative Prompting and Avoiding Undesired Outputs**, 负面提示和避免不期望的输出：探索负面提示和避免大型语言模型产生不期望输出的技术。在其实现中，涵盖基本负面示例、明确排除、使用 LangChain 实现约束，以及评估和改进负面提示词的方法。

2. **Prompt Formatting and Structure**, 提示词格式化和结构：探索各种提示词格式和结构元素，展示它们对 AI 模型响应的影响。在其实现中，演示了创建各种提示词格式、整合结构元素，并比较了不同提示词结构的响应。

3. **Prompts for Specific Tasks**, 用于特定任务的提示词：探索为特定任务创建和使用提示词：文本摘要、问答、代码生成和创意写作。在其实现中，涵盖了设计任务特定的提示词模板，这些模板使用 LangChain 实现，可执行示例输入，并分析每种任务类型的输出。

高级应用

1. **Multilingual and Cross-lingual Prompting**, 多语言和跨语言提示：探索设计在多种语言中有效工作的提示词和语言翻译任务的技术。在其实现中，涵盖创建多语言提示词、实现语言检测和适配、设计跨语言翻译提示词，以及处理各种书写系统和脚本。

2. **Ethical Considerations in Prompt Engineering**, 提示词工程中的伦理考虑：探索提示词工程的伦理维度，重点是避免偏见并创建包容和公平的提示词。在其实现中，涵盖识别提示词中的偏见、实施创建包容性提示词的策略，以及评估和改进 AI 输出伦理质量的方法。

3. **Prompt Security and Safety**，提示词安全性：专注于防止提示词注入和在提示词中实现内容过滤，以确保 AI 应用程序的安全性。其实现中涵盖了提示词注入预防技术、内容过滤实现，以及测试安全措施的有效性。

4. **Evaluating Prompt Effectiveness**，评估提示词效果：探索评估 AI 语言模型中提示词效果的方法和技术。在其实现中，涵盖设置评估指标、实施手动和自动评估技术，并使用 OpenAI 和 LangChain 提供了实践示例。

如果你也正需要一个这样的提示词工程「百科全书」，那就赶紧：

1. 克隆这个库：

```
1 git clone https://github.com/NirDiamant/Prompt_Engineering.git
```

2. 找到你感兴趣的技术：

```
1 cd all_prompt_engineering_techniques
```

3. 跟随每个技术笔记本中的详细实现指南。

当然，你也可以为这个项目添砖加瓦，做出自己的贡献。

参考链接：

https://github.com/NirDiamant/Prompt_Engineering

<https://x.com/NirDiamantAI/status/1844385492801933599>

© THE END

转载请联系本公众号获得授权

投稿或寻求报道：liyazhou@jiqizhixin.com

AI研究前沿 · 目录

上一篇

MetaGPT开源自动生成智能体工作流，
4.55%成本超GPT-4o

下一篇

NeurIPS 2024 (Oral) | 如何量化与提升思维
链的推理能力边界？

留言

都在搜: deepseek提示词万能公式

写留言

