

第一部分 面向开发者的提示工程

Prompt，提示，最初是 NLP 研究者为下游任务设计出来的一种任务专属的输入形式或模板，在 ChatGPT 引发大语言模型新时代之后，Prompt 即成为与大模型交互输入的代称。即我们一般**将给大模型的输入称为 Prompt，将大模型返回的输出称为 Completion。**

随着 ChatGPT 等 LLM（大语言模型）的出现，自然语言处理的范式正在由 Pretrain-Finetune（预训练-微调）向 Prompt Engineering（提示工程）演变。对于具有较强自然语言理解、生成能力，能够实现多样化任务处理的 LLM 来说，一个合理的 Prompt 设计极大地决定了其能力的上限与下限。**Prompt Engineering，即是针对特定任务构造能充分发挥大模型能力的 Prompt 的技巧。**要充分、高效地使用 LLM，Prompt Engineering 是必不可少的技能。

LLM 正在逐步改变人们的生活，而对于开发者，如何基于 LLM 提供的 API 快速、便捷地开发一些具备更强能力、集成 LLM 的应用，来便捷地实现一些更新颖、更实用的能力，是一个急需学习的重要能力。要高效地基于 API 开发集成 LLM 的应用，首要便是学会如何合理、高效地使用 LLM，即如何构建 Prompt Engineering。第一部分 面向开发者的提示工程，源于由吴恩达老师与 OpenAI 合作推出的《ChatGPT Prompt Engineering for Developers》教程，其面向入门 LLM 的开发者，深入浅出地介绍了对于开发者，**如何构造 Prompt 并基于 OpenAI 提供的 API 实现包括总结、推断、转换等多种常用功能**，是入门 LLM 开发的第一步。对于想要入门 LLM 的开发者，你需要充分掌握本部分的 Prompt Engineering 技巧，并能基于上述技巧实现个性化定制功能。

本部分的主要内容包括：书写 Prompt 的原则与技巧；文本总结（如总结用户评论）；文本推断（如情感分类、主题提取）；文本转换（如翻译、自动纠错）；扩展（如书写邮件）等。

目录：

1. 简介 Introduction @邹雨衡
2. Prompt 的构建原则 Guidelines @邹雨衡
3. 如何迭代优化 Prompt Iterative @邹雨衡
4. 文本总结 Summarizing @玉琳
5. 文本推断 @长琴
6. 文本转换 Transforming @玉琳
7. 文本扩展 Expand @邹雨衡
8. 聊天机器人 @长琴
9. 总结 @长琴