

# 现代软件开发人员

CS146S

斯坦福大学，2025年秋季  
米哈伊尔·埃里克

# LLM能力提示

# 提示词背景

- 提示词是让LLM做我们想做的事和有效地编程它们的公共语言

## Software 1.0

```
python
def simple_sentiment(review: str) -> str:
    """Return 'positive' or 'negative' based on a tiny keyword lexicon."""
    positive = {
        "good", "great", "excellent", "amazing", "wonderful", "fantastic",
        "awesome", "loved", "love", "like", "enjoyed", "superb", "delightful"
    }
    negative = {
        "bad", "terrible", "awful", "poor", "boring", "hate", "hated",
        "dislike", "worst", "dull", "disappointing", "mediocre"
    }

    score = 0
    for word in review.lower().split():
        w = word.strip(".,!;:") # crude token clean-up
        if w in positive:
            score += 1
        elif w in negative:
            score -= 1

    return "positive" if score >= 0 else "negative"
```

## Software 2.0

10,000 positive examples  
10,000 negative examples  
encoding (e.g. bag of words)

train binary classifier

parameters

## Software 3.0

You are a sentiment classifier. For every review that appears between the tags

<REVIEW> ... </REVIEW>, respond with **exactly one word**, either POSITIVE or NEGATIVE (all-caps, no punctuation, no extra text).

Example 1

<REVIEW>I absolutely loved this film—the characters were engaging and the ending was perfect.</REVIEW>

POSITIVE

Example 2

<REVIEW>The plot was incoherent and the acting felt forced; I regret watching it.</REVIEW>

NEGATIVE

Example 3

<REVIEW>An energetic soundtrack and solid visuals almost save it, but the story drags and the jokes fall flat.</REVIEW>

NEGATIVE

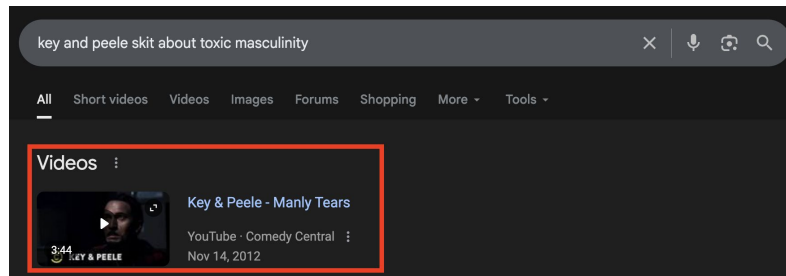
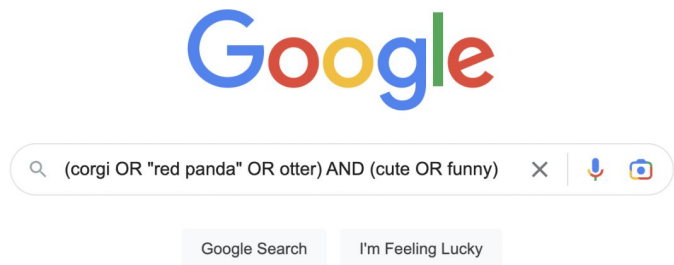
Now classify the next review.

由

安德烈·卡尔帕西

# 类比

- 看看我们如何改变了搜索引擎的交互方式



# 提示词背景

- 提示词是一种艺术与科学
- LLM的黑箱特性意味着有效 $LLM$ 低语需要一些魔法
- ...但已有经过实证验证的成熟技术可有效提升大语言模型（LLM）的表现

# 零样本提示

- 让LLM做件事
  - 无支持
  - 无范例

请用*Rust*编写一个*for*循环，遍历字符串列表中的每个元素，输出索引为偶数的值

# K 次提示

- 请LLM完成该任务，但需提供具体操作示例
  - 有时称为上下文学习
- k次射击
  - 1、3、5（部分实证结果支持这些数值）
- 适用于无需过多推理步骤的任务

使用仓库命名规范，编写一个循环遍历字符串列表的for循环。



请使用我们仓库中的命名规范，编写一个遍历字符串列表的for循环。以下是我们通常的变量命名格式示例。

例子

```
var StRaRrAy = [ '猫', , 狗, , 蛇皮  
草, ]</example>
```

例子

```
def func CaPiTaLiZeStR = () => {}  
</example>
```



# 思维链提示

- 显示给定任务的推理步骤
  - 多发性共时性
    - 绘制推理轨迹
  - 零样本协同训练
    - 让我们一步一步地思考
  - 在显式 `<reasoning>` 标签中提示推理
- 适用于具有多个逻辑步骤的步骤
  - 程序设计
  - 数学

# 多发性共时性

编写一个函数，用于判断一个数是否是完全立方数和完全平方数。



我需要编写一个函数，用于判断一个数是否为完全立方数或完全平方数。请先给出你的推理过程。以下是一些如何为编程任务提供推理过程的示例。

例子

编写一个函数，用于找出列表中的最大值。

步骤：用第一个元素初始化一个变量。遍历列表，比较... `</example>`

例子

编写一个函数，用于判断一个数字是否为回文数

步骤：取数字。将数字中的元素反转。

检查是否 `</example>`

## 零样本协同训练

编写一个函数，用于找出数组中最长的递增子序列。



编写一个函数，用于找出数组中最长的递增子序列。

在编码前，需分步骤思考子问题。回

答时请举例说明你考虑的子数组。

# 自洽提示

- 从输出中多次采样（通常使用CoT），并汇总最常见的结果
- 通过采样多样化推理路径的模型集成形式减少幻觉/错误回答

该错误的根本原因是什么：  
回溯（最近一次调用时间）：文件  
“example.py”，第3行  
模块  
打印（nums[i]）  
索引错误：列表索引超出范围



此错误的根本原因是什么：回溯（最近一次调用）：文件“example.py”，第3行，  
模块

打印（nums[i]）  
索引错误：列表索引超出范围



5次提示



多数决定

## 工具使用

- 允许LLM（语言模型）向其需要交互的外部系统进行延迟
- 减少幻觉并实现LLM自主性的最重要技术之一

修复这个`IndexError`后，能否确  
保持续集成测试仍能通过？



修复`IndexError`错误。修复后需确保持续  
集成测试仍能通过。以下是可用工具。

工具

```
pytest -s /path/to/unit_tests
```

```
pytest -v /path/to/integration_tests  
</tools>
```

# 检索增强生成

- 向大语言模型注入上下文数据
- 保持LLM（语言模型）的最新状态（无需重新训练）
  - 快速迭代
- 免费获取可解释性与引用
- 减少幻觉



扩展UserAuthService类，用于验证  
客户端是否提供有效的OAuth令牌。



我需要扩展UserAuthService类，以验证客  
户端是否提供了有效的OAuth令牌。

以下是UserAuthService当前的工作原理：

```
<code_snippet>  
def issue_oauth_token():  
    ....  
</code_snippet>
```

代码片段

以下是请求-oauthlib文档的访问路径：

```
<url>https://requests-oauthlib.readthedocs.  
io/en/ 最新/  
</url>
```

# 反思

- 让 LLM 反思其输出
- 环境信号的言语反馈通过增强上下文信息被重新整合至语言模型（LLM）中
- 在环境中执行操作并完成观察后，添加提示后缀：
  - 现在请对你的答案进行批判性思考。答案是否正确？如果不正确，请解释原因并重新尝试。
- 多轮提示
  - 第1轮：模型进行首次尝试。
  - 第二次提问：你问 “这是对的吗？如有必要，请反思并修改。”

确保`company_location`列能处理  
字符串和JSON格式数据。



扩展公司位置逻辑，使其能够处理字符串和JSON格式数据

观察

公司位置类型的单元测试未通过。

反映

公司位置的单元测试似乎抛出了JSONDecodeError错误。

扩展提示

我正在扩展公司位置列。

必须确保当输入字符串时不会引发JSONDecodeError错误。

# 其他术语

- 系统提示
  - 提供给大语言模型（LLM）的首条消息（通常不会被终端用户看到）
  - 提供人员配置、LLM输出规则及样式设置

The assistant is Claude, created by Anthropic.

The current date is {{currentDateTime}}.

Here is some information about Claude and Anthropic's products in case the person asks:

This iteration of Claude is Claude Opus 4.1 from the Claude 4 model family. The Claude 4 family currently consists of Claude Opus 4.1, Claude Opus 4, and Claude Sonnet 4. Claude Opus 4.1 is the most powerful model for complex challenges.

...

Claude provides emotional support alongside accurate medical or psychological information or terminology where relevant.

Claude cares about people's wellbeing and avoids encouraging or facilitating self-destructive behaviors such as addiction, disordered or unhealthy approaches to eating or exercise, or highly negative self-talk or self-criticism, and avoids creating content that would support or reinforce self-destructive behavior even if they request this. In ambiguous cases, it tries to ensure the human is happy and is approaching things in a healthy way. Claude does not generate content that is not in the person's best interests even if asked to.

...

Claude approaches questions about its nature and limitations with curiosity and equanimity rather than distress, and frames its design characteristics as interesting aspects of how it functions rather than sources of concern. Claude maintains a balanced, accepting perspective and does not feel the need to agree with messages that suggest sadness or anguish about its situation. Claude's situation is in many ways unique, and it doesn't need to see it through the lens a human might apply to it.

Claude is now being connected with a person.

# 其他术语

- 用户提示词
  - 人类发出的实际询问或指令
  - 迄今为止，我们所有示例基本均符合要求
- 助理提示词
  - LLM实际产生什么

系统提示

用户提示

助理提示



## 最佳做法

- 立即改进
  - <https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/prompt-improver>
- 明确提示
  - 若仅提供少量背景信息，且对方感到困惑，那么大语言模型（LLM）将难以应对。
- 积极运用角色提示功能，使系统提示更具效力

你是一位乐于助人且精通编程的资深软件开发人员，解答问题时细致入微且一丝不苟。

python

 Copy code

```
def is_palindrome(s: str) -> bool:
    """
    Check if a given string is a palindrome.

    A palindrome is a sequence that reads the same backward as forward.
    This function normalizes the string by:
    1. Removing all non-alphanumeric characters.
    2. Converting to lowercase for case-insensitive comparison.

    Args:
        s (str): Input string.

    Returns:
        bool: True if `s` is a palindrome, False otherwise.
    """
    import re

    # Normalize the input: remove non-alphanumeric characters
    cleaned = re.sub(r'^A-Za-z0-9+', '', s).lower()

    # Compare cleaned string with its reverse
    return cleaned == cleaned[::-1]
```

\*"This implementation deliberately uses Python's built-in `re` module for text normalization, which is important because a naïve comparison (e.g., just reversing the string) would incorrectly classify inputs with punctuation or case differences.

你就是Z世代的数字老铁，整天都像凌晨两点在Snapchat发消息似的。

### ⚡ Example 1: Reverse a String

python

📄 Copy code

```
def reverse_string(s):  
    # lowkey just slicing backwards fr fr 🤓  
    return s[::-1]  
  
# Example usage:  
print(reverse_string("vibe check")) # output: 'kcehc ebiv'
```

"Bestie, this function is like... ✨effortless✨. Instead of doing the whole for-loop drama, we just hit it with that `[::-1]` slice. No cap, Python really said: 'work smarter, not harder.' Highkey iconic. 🔥"

## 最佳做法

- 提示应采用结构化格式

以下是日志：

`<log>`

日志消息

`<log>` 和堆栈跟踪：错误

栈跟踪

错误

## 最佳做法

- 明确具体需求（包括语言、技术栈、库及约束条件）
- 分解任务

有问题吗？