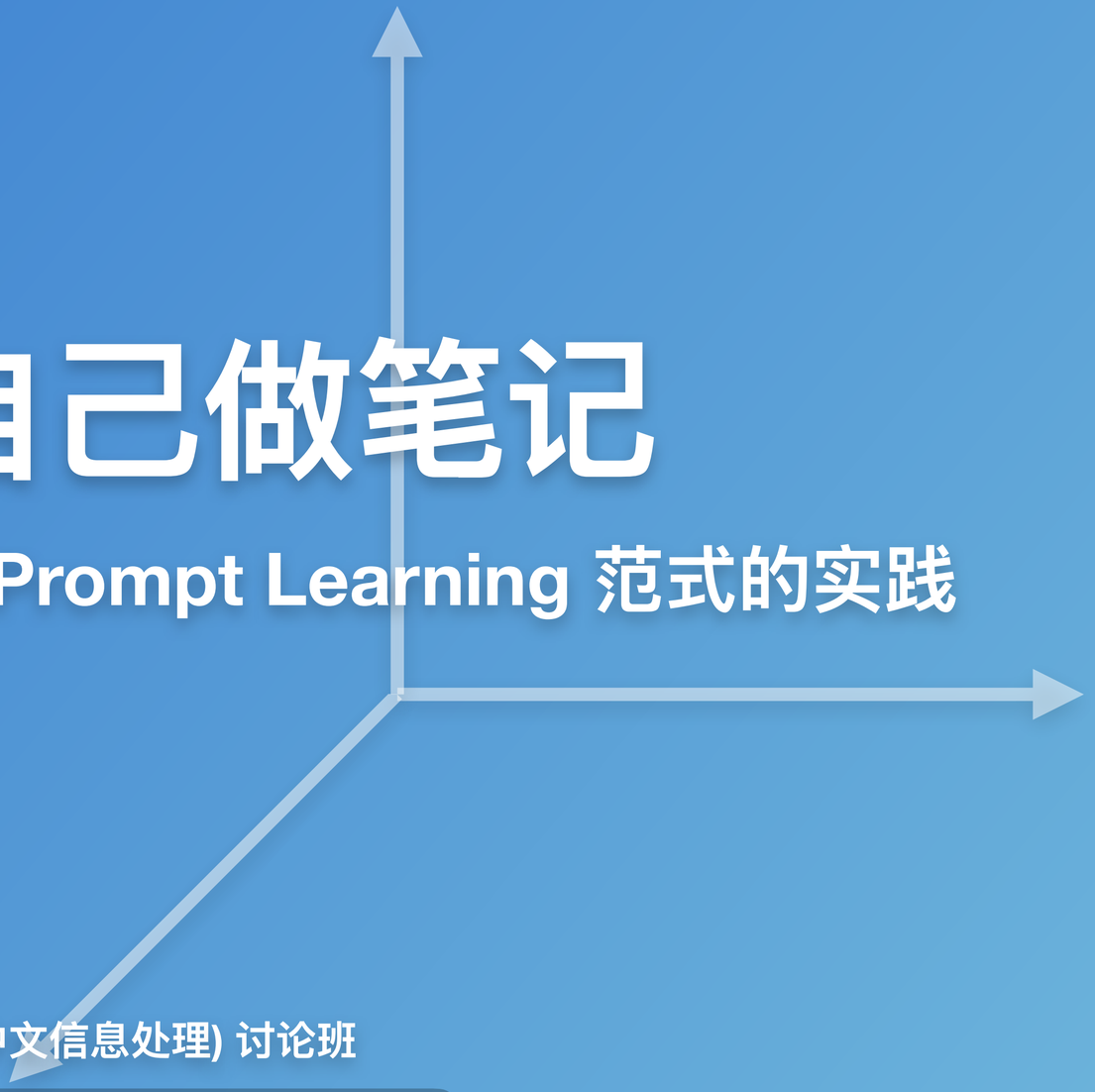
SWOT：让 LLM 自己做笔记

一种 Self-Prompt Training / System Prompt Learning 范式的实践



孙春晖　2025年05月29日　北京大学　现代汉语 (计算语言学 | 中文信息处理) 讨论班

立即访问 https://github.com/ziioai/swot 查看代码和 Demo

目录

SWOT：让 LLM 自己做笔记

一种 Self-Prompt Training / System Prompt Learning 范式的实践

本次报告侧重于工程技术和工具使用方面的经验分享

引子

引子：SpaCE2025 第一名的做法

【得分：zh_0.8686；en_0.8251】

【第二名得分：zh_0.6254；en_0.5997】

4.1 任务提示设计

对于空间理解任务，我们构建了包含以下要素的详细框架：

Prompt

你是一名空间关系分析专家，请严格按步骤执行：1. 识别文本中所有空间描述（位置/方向/移动）2. 检查是否存在以下问题：- 矛盾词组合（如”没入水上”应改为”没入水下”）- 趋向动词与介词搭配错误（如”回在床边”应改为”回到床边”）- 物理常识错误（如”后视镜看前方”应改为”后视镜看后方”）- 运动方向矛盾（如”向下形成Y字”应改为”向上形成Y字”）- 空间参照实体错误（如”绕到外面来”应改为”绕到前面来”）3. 车辆转向方向判定应该根据车辆行驶方向及车头朝向判断左右转向的实际方位（例：车头朝西行驶等价于由东向西行驶，此时左转进入南向道路，右转进入北向道路；由东向北行驶就是指右转）4. 如果发现明显与常识或空间认知矛盾即标记【错误】，否则【正确】
请判断以下文本的空间表达，回答【正确】或【错误】：

Prompt

请对比text1和text2中描述物体位置的关键空间关系（如上下、内外、前后等方位词，以及物体放置位置等）。若文本差异仅改变观察视角或描述方式，但物体间的实际空间关系保持一致（如”树下”和”树前”都表示靠近树的位置，为空间近义词），又或者文本差异之间属于包含关系，则判断为【相同】，若文本差异矛盾或对立，动作空间关系相反（如”进去”和”出去”为空间反义词），物体间的实际空间关系相反（如”树下”和”树上”为空间反义词），则判断为【不同】。

Prompt

请根据以下步骤分析文本并回答问题：
1. **定位关键方位词**：找出句子中提到的方位词（如“侧面”、“上面”、“对面”等）。2. **确定上下文主体**：明确方位词修饰的主体（如人物、物体）及其位置关系。3. **识别基准对象**：根据上下文推断方位词的基准（通常为动作的主体、最近的物体或明确提及的参照物）。4. **验证解释**：对比interpretation中的基准是否与文本推断一致。若一致则回答【正确】，否则【错误】。
示例分析：- 文本：“李国秀在浇花，张顺东在其后方拍摄，陆金云指导‘绕到侧面’。” - 关键方位词：“侧面”。- 上下文主体：张顺东正在拍摄李国秀。- 基准推断：“侧面”应以李国秀的位置为基准。- 验证：若interpretation为“以李国秀为基准”，则答案【正确】。
请按此逻辑判断以下问题：

对于SPR类推理任务，我们构建了包含以下要素的详细框架：

空间坐标系统定义：为每种空间布局建立精确的坐标系方位关系数学化表示：使用向量和角度精确定义空间关系推理算法指导：提供向量叉乘法等具体计算方法特殊情况处理规则：针对复杂空间关系的判断标准

Prompt

请解决以下座位布局推理题。四个人坐在长方形桌子的两侧，如下图所示：上排— 甲座（右侧）— 乙座（左侧）—-———-——- 下排— 丙座（左侧）— 丁座（右侧）
其中，上排的人面向下方，下排的人面向上方。”左”和”右”需要根据每个人的朝向来判断的，上排的左侧和右侧与下排的左侧和右侧相反。
已知：y1，位于上排座位；y2，位于下排座位。
问题：item[”question”]
选项：options
甲座的正对面是丙座，乙座的正对面是丁座。
斜对面：指同侧相邻人的正对面。甲座的斜对面是丁座，乙座的斜对面是丙座。
右前方：指右边人的正对面。左前方指左边人的正对面。甲座的左前方是丁座，丁座的左前方是甲座；乙座的右前方是丙座，丙座的右前方是丁座。
X在Y的某方向：指的是以Y为参照系，Y的朝向为正北方向，然后确定X在Y的某方向。
题目是单选题，只有一个正确答案，请将最终答案放入□

4.2 批量处理与多数投票

为了提高推理的稳定性和准确性，我们实现了以下机制：

异步处理：支持多GPU并行推理实时写入：即时保存推理结果防止数据丢失对每个问题生成多个推理结果提取标准化答案格式（使用正则表达式）基于频次统计确定最终答案区分单选题和多选题的投票策略

引子：SpaCE2025 第一名的做法

张天师、汉钟离、姜子牙、张果老四人来到火锅店吃火锅，选了四人卡座坐下。卡座分列一张长方形桌子长边两侧，每排卡座上坐两人。面对面而坐。已知：
(1)张果老是姜子牙的左邻；
(2)汉钟离在张天师同侧右边。

请解决以下座位布局推理题。

张天师、汉钟离、姜子牙、张果老四人来到火锅店吃火锅，选了四人卡座坐下。卡座分列一张长方形桌子长边两侧，每排卡座上坐两人。面对面而坐。

四个人坐在长方形桌子的两侧，如下图所示：

上排	甲座（右侧）	乙座（左侧）
下排	丙座（左侧）	丁座（右侧）

其中，上排的人面向下方，下排的人面向上方。”左"和"右"需要根据每个人的朝向来判断的，上排的左侧和右侧与下排的左侧和右侧相反。

已知：**(1)张果老是姜子牙的左邻**，位于上排座位；**(2)汉钟离在张天师同侧右边**，位于下排座位。

问题：{item["question"]}
选项：
{options}

甲座的正对面是丙座，乙座的正对面是丁座。
"斜对面"：指同侧相邻人的正对面。甲座的斜对面是丁座，乙座的斜对面是丙座。
"右前方"：指右边人的正对面。"左前方"指左边人的正对面。甲座的左前方是丁座，丁座的左前方是甲座；乙座的右前方是丙座，丙座的右前方是丁座。
X在Y的某方向：指的是以Y为参照系，Y的朝向为正北方向，然后确定X在Y的某方向。

他们人工地为题目补充了很多信息

引子：SpaCE2025 第一名的做法

```python
def change_text_zh(item):
    if "四人来到火锅店吃火锅" in item["text"]:
        # 350条
        pass

    elif "六盆花放置在三层花架上呈列" in item["text"]:
        # 350条
        item["text"] = item["text"].replace("花架从下至上依次为第一、二、三层", "花架从上至下依次为第三、二、一层")

    elif "终南山重阳宫内盘腿席地打坐，围成一个圆圈" in item["text"]:
        # 635条
        item["text"] = (
            item["text"]
            .replace(
                "六位道士在终南山重阳宫内盘腿席地打坐，围成一个圆圈，修炼内功，六人的位置恰好形成一个正六边形。",
                "六位道士在终南山重阳宫内围成一个圆圈，六人的位置恰好落在正六边形的六个顶点上。",
            )
            .replace("任意相邻两人之间的间距相等，大约为一米。", "")
        )

        if "六人都面朝外背对圆心而坐" in item["text"]:
            item["text"] = item["text"].replace("六人都面朝外背对圆心而坐", "六人都面朝外“背对”圆心而坐")
        if "六人都面朝内正对圆心而坐" in item["text"]:
            item["text"] = item["text"].replace("六人都面朝内正对圆心而坐", "六人都面朝内“正对”圆心而坐")

    elif "六座神像在神坛中围成一个圆圈" in item["text"]:
        # 665条
        item["text"] = item["text"].replace("任意相邻两个神像之间的距离相等，大约为一米。", "")

        if "每座神像都背对神坛中心" in item["text"]:
            item["text"] = item["text"].replace("每座神像都背对神坛中心", "每座神像都“背对”神坛中心")
        if "每座神像都正对神坛中心" in item["text"]:
            item["text"] = item["text"].replace("每座神像都正对神坛中心", "每座神像都“正对”神坛中心")

    return item
```

谭处端、丘处机、赵志敬、柯镇恶、王处一、郝大通六位道士在终南山重阳宫内盘腿席地打坐，围成一个圆圈，修炼内功，六人的位置恰好形成一个正六边形。六人都面朝外背对圆心而坐。任意相邻两人之间的间距相等，大约为一米。六人朝向分别是正东、正西、西北、东北、西南、东南。已知：
(1)谭处端的左边紧接着就是丘处机；
(2)赵志敬朝西北方向；
(3)丘处机背对的位置是郝大通右边第一个位置；
(4)赵志敬在柯镇恶的右边，二者相邻；
(5)谭处端与柯镇恶正背对；
(6)王处一在柯镇恶左侧紧邻位置。

谭处端、丘处机、赵志敬、柯镇恶、王处一、郝大通六位道士在终南山重阳宫内围成一个圆圈，六人的位置恰好**落在正六边形的六个顶点上**。六人都面朝外背对圆心而坐。六人朝向分别是正东、正西、西北、东北、西南、东南。已知：
(略)

他们人工地为修改了题目的表述，使其更加精炼

引子：SpaCE2025 第一名的做法

请解决以下空间方位/布局推理单项选择题，该题只有一个正确选项：

谭处端、丘处机、赵志敬、柯镇恶、王处一、郝大通六位道士在终南山重阳宫内围成一个圆圈，六人的位置恰好**落在正六边形的六个顶点上**。六人都面朝外背对圆心而坐。六人朝向分别是正东、正西、西北、东北、西南、东南。已知：
(1)谭处端的左边紧接着就是丘处机；
(2)赵志敬朝西北方向；
(3)丘处机背对的位置是郝大通右边第一个位置；
(4)赵志敬在柯镇恶的右边，二者相邻；
(5)谭处端与柯镇恶正背对；
(6)王处一在柯镇恶左侧紧邻位置。

问题：{item["question"]}
选项：
{options}

解题须知：

1. 以正六边形中心为原点，每个顶点间隔60度，六个顶点的全局坐标按顺时针排列：

顶点编号	坐标 (x, y)	全局方向
1	(1, 0)	正东
2	(–0.5, √3/2)	东南
3	(–0.5, –√3/2)	西南
4	(-1, 0)	正西
5	(0.5, –√3/2)	西北
6	(0.5, √3/2)	东北

2. “左边”“右边”定义
   - 均基于每个实体自身的参照系。当描述“A在B的左边”时，这是从B的视角出发，以B为参照点，B的面朝方向为正北，此时A位于B的西侧。
   - 对于任意一个顶点，其左边第一个顶点和左边第二个顶点均属于其左侧；其右边第一个顶点和右边第二个顶点均属于其右侧。
   - 甲的“斜对面”指甲正对面（相隔3个顶点）两侧相邻的那两个实体。

3. 顺/逆时针等价转换
   - “A在B左（右）边第n个位置” ≡ “A在B右（左）边第(6–n)个位置”。
   - “A在B顺时针第n个位置” ≡ “A在B逆时针第(6–n)个位置”。

4. 斜对面
   - A的“斜对面”指A正对面（相隔3个顶点）两侧相邻的那两个实体。

5. “()背对的位置”
   - “甲背对的位置是乙” ≡ “甲和乙背对圈中心，甲的正对面/正后方是乙”。

6. “X朝着某方向”
   - 例如“X朝西南”即X面向全局参照系的西南方向。
   - 若X面向圈心，则X的左手边实体面朝正南，右手边实体面朝西北；
   - 若X背向圈心，则X的左手边实体面朝西北，右手边实体面朝正南；

7. 绝对方位判断
   - 设A、B的全局坐标分别为(x_A,y_A)、(x_B,y_B)：
     - “A在B的东北边” ⇔ x_A > x_B 且 y_A > y_B
     - “东南” ⇔ x_A > x_B 且 y_A < y_B
     - “西南” ⇔ x_A < x_B 且 y_A < y_B
     - “西北” ⇔ x_A < x_B 且 y_A > y_B

8. 向量叉乘法（通用左右/前后判断）
   对任意已定位置的参考者A和目标B：
   a. 取正六边形中心O为原点，OA, OB为对应坐标向量；
   b. 令A的面朝向向量v =
      { +OA/|OA|　若A背离圈心
      { –OA/|OA|　若A面向圈心
   c. 令w = (x_B–x_A, y_B–y_A)；
   d. 计算二维叉乘标量
      $z = v_x \cdot w_y - v_y \cdot w_x$
     - 若 z > 0，则B在A的左侧
     - 若 z < 0，则B在A的右侧
     - 若 z = 0，则B在A的正前方或正后方

9. 解题步骤
   a. 确定每人“面向圈心”或“背离圈心”；
   b. 确定每人的顶点坐标；
   c. 使用合适的方法解答问题
   d. 该问题为单选题，分析问题并选出1个正确选项放入一个单独\boxed{}

1 相关背景

System Prompt Learning 范式的提出

System Prompt Learning 范式的提出

- Andrej Karpathy（OpenAI创始成员之一）于 5月11日在 X 上就 Claude 长达近17000词的系统提示词 发表看法：
- “我们似乎遗漏了一种重要的LLM学习范式”，或可称为“系统提示学习”。

Claude 系统提示词的构成

https://x.com/karpathy/status/1921368644069765486

https://www.dbreunig.com/2025/05/07/claude-s-system-prompt-chatbots-are-more-than-just-models.html

We're missing (at least one) major paradigm for LLM learning. Not sure what to call it, possibly it has a name - system prompt learning?

Pretraining is for knowledge.
Finetuning (SL/RL) is for habitual behavior.

Both of these involve a change in parameters but a lot of human learning feels more like a change in system prompt. You encounter a problem, figure something out, then "remember" something in fairly explicit terms for the next time. E.g. "It seems when I encounter this and that kind of a problem, I should try this and that kind of an approach/solution". It feels more like taking notes for yourself, i.e. something like the "Memory" feature but not to store per-user random facts, but general/global problem solving knowledge and strategies. LLMs are quite literally like the guy in Memento, except we haven't given them their scratchpad yet. Note that this paradigm is also significantly more powerful and data efficient because a knowledge-guided "review" stage is a significantly higher dimensional feedback channel than a reward scaler.

I was prompted to jot down this shower of thoughts after reading through Claude's system prompt, which currently seems to be around 17,000 words, specifying not just basic behavior style/preferences (e.g. refuse various requests related to song lyrics) but also a large amount of general problem solving strategies, e.g.:

"If Claude is asked to count words, letters, and characters, it thinks step by step before answering the person. It explicitly counts the words, letters, or characters by assigning a number to each. It only answers the person once it has performed this explicit counting step."

This is to help Claude solve 'r' in strawberry etc. Imo this is not the kind of problem solving knowledge that should be baked into weights via Reinforcement Learning, or least not immediately/exclusively. And it certainly shouldn't come from human engineers writing system prompts by hand. It should come from System Prompt learning, which resembles RL in the setup, with the exception of the learning algorithm (edits vs gradient descent). A large section of the LLM system prompt could be written via system prompt learning, it would look a bit like the LLM writing a book for itself on how to solve problems. If this works it would be a new/powerful learning paradigm. With a lot of details left to figure out (how do the edits work? can/should you learn the edit system? how do you gradually move knowledge from the explicit system text to habitual weights, as humans seem to do? etc.).

8:55 AM · May 11, 2025 · **1.3M** Views

我们似乎遗漏了（至少一种）重要的LLM学习范式。我不确定该如何命名它，或许已有专业术语——系统提示学习？

预训练是为了获取知识，
微调（监督学习/强化学习）是为了培养习惯性行为。

这两种方式都涉及参数调整，但人类的大量学习过程更像是在更新系统提示。当你遇到问题、找到解决方案后，会以非常明确的方式"记住"应对策略。比如"当遇到这类问题时，我应该尝试那种解决方法"。这更像是为自己做笔记，类似于"记忆"功能，但不是用来存储用户随机信息，而是记录通用的问题解决知识和策略。当前的LLM就像《记忆碎片》里的主角，只不过我们还没给它们准备记事本。值得注意的是，这种范式在数据效率上更为强大，因为知识引导的"复盘"阶段提供的反馈维度，远比单一的奖励标量丰富得多。

促使我写下这些思考的契机，是阅读了Claude长达约17,000词的系统提示。这些提示不仅规定了基础行为风格（比如拒绝与歌词相关的各种请求），还包含了大量通用问题解决策略，例如：

"当被要求统计单词、字母或字符时，Claude会在回答前逐步思考。它会明确地为每个元素编号统计，只有在完成这个显式计数步骤后才会给出答案。"

这种设计是为了解决类似"草莓单词中有几个r"的问题。在我看来，这类问题解决知识不应该通过强化学习固化到权重中（至少不应该立即或完全依赖这种方式），更不应该依赖工程师手工编写系统提示。它应该来自系统提示学习——这种模式在架构上类似强化学习，区别在于学习算法（直接编辑vs梯度下降）。LLM系统提示的很大部分内容都可以通过这种学习方式生成，就像是LLM为自己编写的问题解决指南。如果可行，这将形成一种全新且强大的学习范式。当然还有大量细节需要探索：编辑机制如何运作？能否/应该学习编辑系统？如何像人类那样将显性知识逐步转化为习惯性权重？等等。

System Prompt Learning 范式的提出

存在一类问题

Strawberry
中有几个“r”?

Claude 的策略

人为地在 System Prompt 中写下：

"当被要求统计单词、字母或字符时，Claude会在回答前逐步思考。它会明确地为每个元素编号统计，只有在完成这个显式计数步骤后才会给出答案。"

Karpathy 的反思

这种写在 System Prompt 中的**针对特定问题的策略**，是不是应该让模型自己来生成?

对比

预训练 → 获取知识
微调 → 形成习惯行为

SPL → 记下针对性的策略笔记

LLM 对话程序的工作原理

LLM 对话程序的工作原理

User

你好

Assistent

你好！😊 很高兴见到你，有什么我可以帮你的吗?

这是我们看到的界面

LLM 对话程序的工作原理

System

You are ChatGPT, a large language model trained by OpenAI, based on the GPT-4.5 architecture.

Knowledge cutoff: 2023-10

Current date: {CURRENT_DATE}

(……)

User

你好

Assistent

你好！😊 很高兴见到你，有什么我可以帮你的吗?

通常，系统提示词在界面上是隐藏的。通过训练或工程手段，系统提示词被赋予了更高的注意力权重。
此页系统提示词来自 https://github.com/asgeirtj/system_prompts_leaks/tree/main

LLM 对话程序的工作原理

System

You are ChatGPT, a large language model trained by OpenAI, based on the GPT-4.5 architecture.

Knowledge cutoff: 2023-10

Current date: {CURRENT_DATE}

(……)

User

你好

Assistent

你好！😊 很高兴见到你，有什么我
可以帮你的吗?

排除界面的干扰，对话和文章没什么区别，都是连续文本

LLM 对话程序的工作原理

<|system|> You are ChatGPT, a large language model trained by OpenAI, based on the GPT-4.5 architecture.

Knowledge cutoff: 2023-10

Current date: {CURRENT_DATE}

(……) <|endoftext|>

注意此页这些特殊标记可能并不符合真实情况，仅用于演示

<|user|> 你好 <|endoftext|>

<|assistant|> 你好！😊 很高兴见到你，有什么我 可以帮你的吗？ <|endoftext|>

模型的训练语料会用一些特殊 token 来标记不同片段类型

参考 https://github.com/kitools/deepseek-token-counter/blob/main/public/deepseek_v3_tokenizer/tokenizer_config.json

LLM 对话程序的工作原理

<|system|> You are ChatGPT, a large language model trained by OpenAI, based on the GPT-4.5 architecture.

Knowledge cutoff: 2023-10

Current date: {CURRENT_DATE}

(……) <|endoftext|>

<|user|> 你好 <|endoftext|>

<|assistant|> 你好！😊 很高兴见到你，有什么我 可以帮你的吗？ <|endoftext|>

LLM 对话程序的工作原理

<|system|> You are ChatGPT, a large language model trained by OpenAI, based on the GPT-4.5 architecture.

Knowledge cutoff: 2023-10

Current date: {CURRENT_DATE}

(……) <|endoftext|>

<|user|> 你好 <|endoftext|>

<|assistant|> 你好！😊 很高兴见到你，有什么我 可以帮你的吗？ <|endoftext|>

<|user|> 你是谁 <|endoftext|>

<|assistant|>

用户发送消息，相当于人工续写了一部分文本
模型就是在此基础上续写

LLM 对话程序的工作原理

<|system|> You are ChatGPT, a large language model trained by OpenAI, based on the GPT-4.5 architecture.

Knowledge cutoff: 2023-10

Current date: {CURRENT_DATE}

(……) <|endoftext|>

<|user|> 你好 <|endoftext|>

<|assistant|> 你好！😊 很高兴见到你，有什么我 可以帮你的吗？ <|endoftext|>

<|user|> 你是谁 <|endoftext|>

<|assistant|>我是 ChatGPT，由 OpenAI 公司开发的智能AI助手！✨ 我的使命是为你提供各种帮助，无论是解答问题、提供建议、陪你聊天，还是

如果开启流式传输，那么每个 token 都会发送给前端

LLM 对话程序的工作原理

<|system|> You are ChatGPT, a large language model trained by OpenAI, based on the GPT-4.5 architecture.

Knowledge cutoff: 2023-10

Current date: {CURRENT_DATE}

(……) <|endoftext|>

<|user|> 你好 <|endoftext|>

<|assistant|> 你好！😊 很高兴见到你，有什么我 可以帮你的吗？ <|endoftext|>

<|user|> 你是谁 <|endoftext|>

<|assistant|>我是 ChatGPT，由 OpenAI 公司开发的智能AI助手！✨ 我的使命是为你提供各种帮助，无论是解答问题、提供建议、陪你聊天，还是帮你查找信息，我都会尽力做到最好。😊

有什么想问的，或者需要帮忙的吗？随时告诉我哦！**<|endoftext|>**

程序检测到特殊标记，停止续写

LLM 对话程序的工作原理

要点：

1、本质上是文本续写，Next Token Prediction

2、系统提示词被赋予了特殊权重

3、续写的终止是由特殊 token 触发的

4、续写会受到完整的整个上下文的影响

Function Call / Tool Using 的工作原理

Function Call / Tool Using 的工作原理

<|system|> You are ChatGPT (……)

(……中间含有一些工具接口说明……)

<|endoftext|>

<|user|> 计算 235*642 等于多少 <|endoftext|>

<|assistant|> 我将使用计算器来计算， **<|tool_calls_begin|>** function
<|tool_sep|> calculator
```json
{ “exp”: “235 * 642” }
```**<|tool_calls_end|>**

其实就是引入新的特殊 token ，触发程序调用工具函数，
此时程序让模型暂停续写

Function Call / Tool Using 的工作原理

<|system|> You are ChatGPT (……)

(……中间含有一些工具接口说明……)

<|endoftext|>

<|user|> 计算 235*642 等于多少 <|endoftext|>

<|assistant|> 我将使用计算器来计算， <|tool_calls_begin|> function
<|tool_sep|> **calculator**
```json
**{ “exp”: “235 * 642” }**
```<|tool_calls_end|>

<|tool_outputs_begin|> { “result”: 150870 } <|tool_outputs_end|>

程序根据函数的名称和参数，调用相应函数，并把结果按照特定格式插入到文本中

Function Call / Tool Using 的工作原理

<|system|> You are ChatGPT (……)

(……中间含有一些工具接口说明……)

<|endoftext|>

<|user|> 计算 235*642 等于多少 <|endoftext|>

<|assistant|> 我将使用计算器来计算， <|tool_calls_begin|> function
<|tool_sep|> calculator
```json
{ “exp”: “235 * 642” }
```<|tool_calls_end|>

<|tool_outputs_begin|> { “result”: 150870 } <|tool_outputs_end|>

结果是 150870 。<|endoftext|>

然后模型继续续写，直到输出结束符号，触发停止

Function Call / Tool Using 的工作原理

User

计算 235*642 等于多少

Assistent

我将使用计算器来计算，

已使用 calculator

结果是 150870 。

用户界面上可能会渲染成类似这样

Function Call / Tool Using 的工作原理

要点：

1、由特殊 token 触发工具的调用

2、程序将工具函数的返回结果插入上下文

3、模型做的事情仍然始终都是文本续写

RAG 相关技术及其原理

检索增强生成（Retrieval Augmented Generation, RAG）

RAG 相关技术及其原理

朴素RAG，这种方法通常是作为所有现有RAG系统的标准基线。它首先将输入文档分割成几个文本块，并利用文本嵌入将它们编码进向量空间。然后基于查询表示的相似性检索相关文本块。

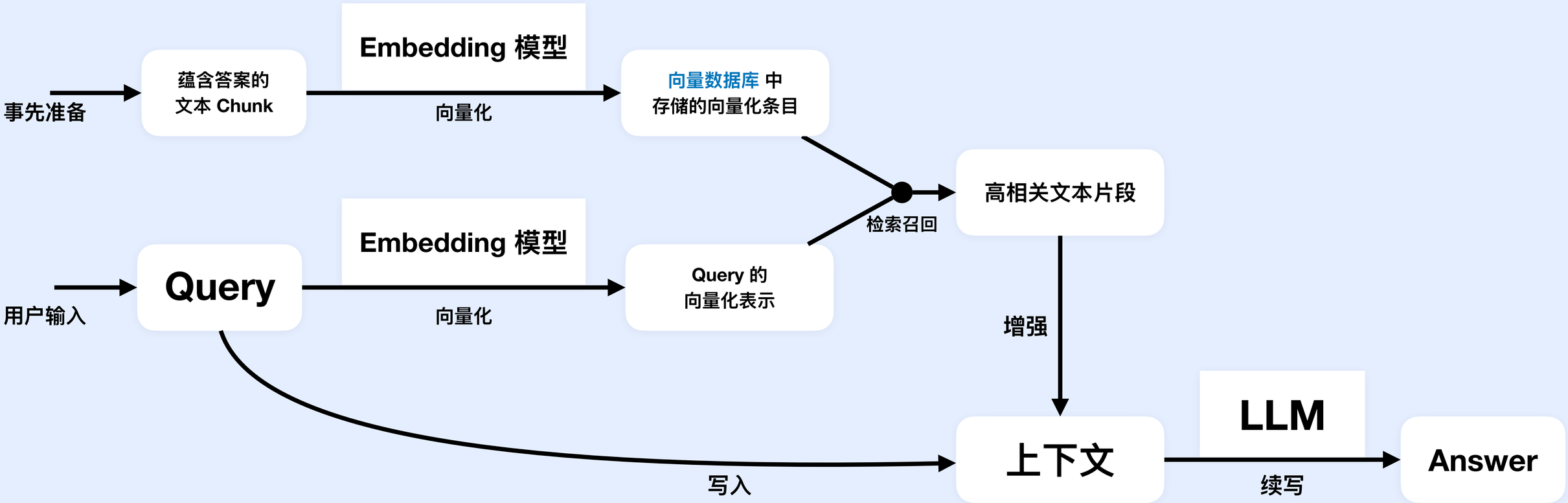
Hyde作为传统RAG系统的改进方法，首先生成“假设性”的文本以捕捉查询的本质。然后使用生成的文本来从大型语料库中检索相关文档，在嵌入空间中使用向量相似性。该方法在前端修改输入查询，而不改变文本块或其嵌入。

GraphRAG首先将输入文本分割成块，并提取其中的实体和关系，形成图结构。然后将这个图划分为多个不同级别的社区。在查询时，GraphRAG识别出问题中的相关实体，并通过参考这些相应的社区摘要来合成答案。与传统的RAG方法相比，GraphRAG提供了对整个文档更结构化、更高层次的理解。

LightRAG是一种基于GraphRAG的改进方法，旨在通过双层检索在最小化计算开销的同时增强检索信息的全面性。这导致检索效率更高，并且与GraphRAG相比，在效果和速度之间实现了更好的平衡。

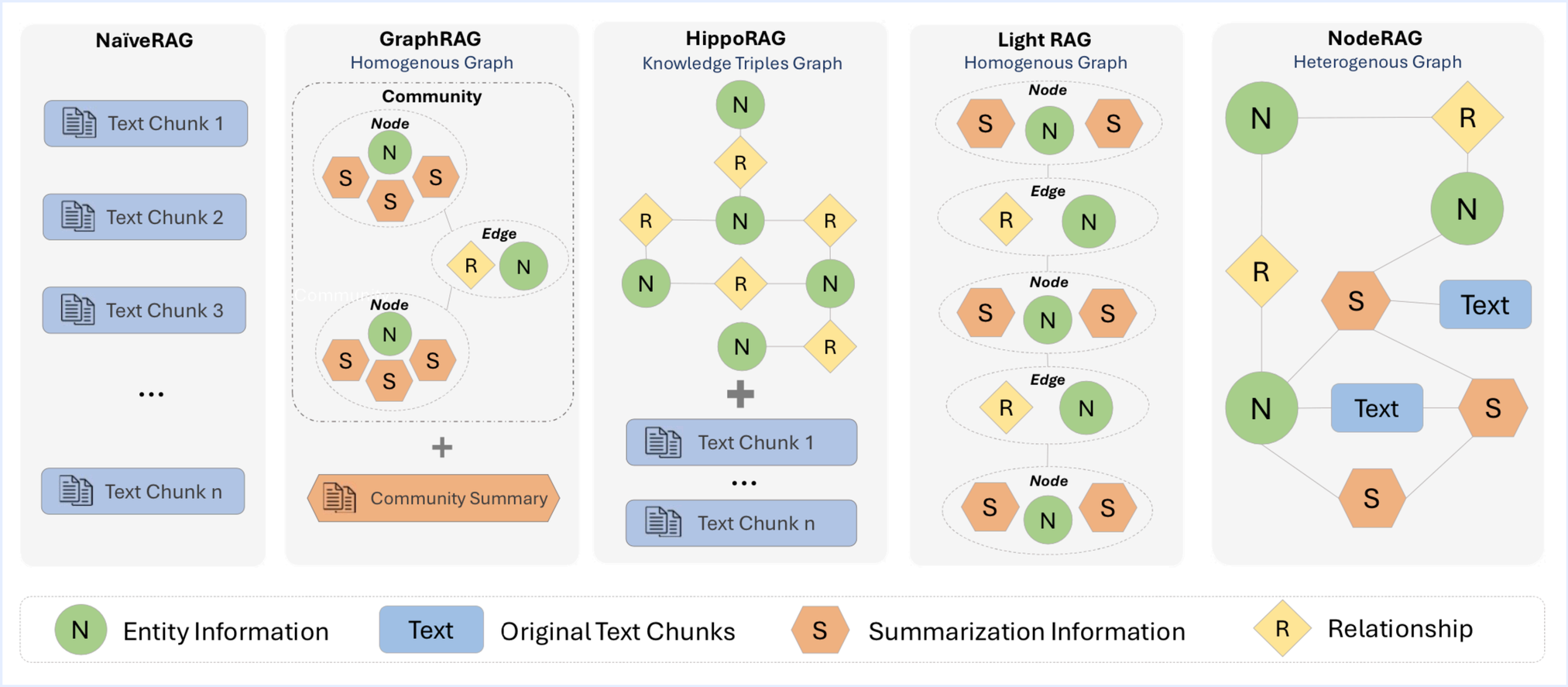
参考 老刘说NLP NodeRAG实现思路解读 https://mp.weixin.qq.com/s/b0lJXysRC767Pmt5EFrsHg

朴素的 NaiveRAG 的工作原理



参考 大模型 RAG 技术概览 https://zhuanlan.zhihu.com/p/678893732

RAG 相关技术及其原理



NaïveRAG 检索的是碎片化的文本片段，这会导致信息重复；HippoRAG 引入了知识图谱，但缺乏对高级别的总结；
GraphRAG 检索的是社区总结，但仍然可能产生粗粒度的信息；LightRAG 结合了一跳邻居，但检索出的节点存在冗余。
这些方法导致检索方法中的不一致性（分离局部和全局检索），并导致粗粒度检索，即不加区分地检索一个实体包括所有相关内容。
而 NodeRAG 利用多种节点类型，包括高级元素、语义单元和关系，从而能够实现更精确、层次化的检索，同时减少无关信息。

参考 NodeRAG 论文 https://arxiv.org/abs/2504.11544
参考 老刘说NLP NodeRAG实现思路解读 https://mp.weixin.qq.com/s/b0lJXysRC767Pmt5EFrsHg

RAG 的其他变体

KAG

KAG 是基于 OpenSPG 引擎和大型语言模型的逻辑推理问答框架，用于构建垂直领域知识库的逻辑推理问答解决方案。KAG 可以有效克服传统 RAG 向量相似度计算的歧义性和 OpenIE 引入的 GraphRAG 的噪声问题。KAG 支持逻辑推理、多跳事实问答等，并且明显优于目前的 SOTA 方法。

KAG 的目标是在专业领域构建知识增强的 LLM 服务框架，支持逻辑推理、事实问答等。KAG 充分融合了 KG 的逻辑性和事实性特点，其核心功能包括：

- 知识与 Chunk 互索引结构，以整合更丰富的上下文文本信息
- 利用概念语义推理进行知识对齐，缓解 OpenIE 引入的噪音问题
- 支持 Schema-Constraint 知识构建，支持领域专家知识的表示与构建
- 逻辑符号引导的混合推理与检索，实现逻辑推理和多跳推理问答

https://github.com/OpenSPG/KAG/blob/master/README_cn.md

LazyGraphRAG

LazyGraphRAG的一个关键优势是**其在成本和质量方面的固有可扩展性**。在一系列竞争方法（**标准向量RAG、RAPTOR、GraphRAG本地、GraphRAG全局和DRIFT搜索机制**）中，LazyGraphRAG在成本–质量上显示出强大的性能，如下所示：

- LazyGraphRAG数据索引成本与向量RAG相同，是完整GraphRAG成本的0.1%。
- 对于与向量RAG相当的查询成本，LazyGraphRAG在本地查询上超越了所有竞争方法，包括长上下文向量RAG和GraphRAG DRIFT搜索（我们最近引入的RAG方法，已被证明优于向量RAG）以及GraphRAG本地搜索。
- 相同的LazyGraphRAG配置还显示出与GraphRAG全局搜索相当的答题质量，但对于全局查询，**查询成本降低了700多倍**。
- 对于GraphRAG全局搜索查询成本的4%，LazyGraphRAG在本地和全局查询类型上显著超越了所有竞争方法，包括C2级别的GraphRAG全局搜索（社区层级中的第三层，推荐大多数应用使用）。

https://www.microsoft.com/en-us/research/blog/lazygraphrag-setting-a-new-standard-for-quality-and-cost/

Nano-GraphRAG 一个精简的 GraphRAG 实现

https://github.com/gusye1234/nano-graphrag

RAG 相关技术及其原理

要点：

1、主要是为了解决知识（或者说“信息”）的召回问题

2、常使用向量化表示和图结构的组织形式

2 SWOT 系统介绍

SWOT 系统概述及功能演示

请看实际演示 可同步查看 Demo 网址 https://ziioai.github.io/swot

SWOT 系统概述及功能演示

要点：

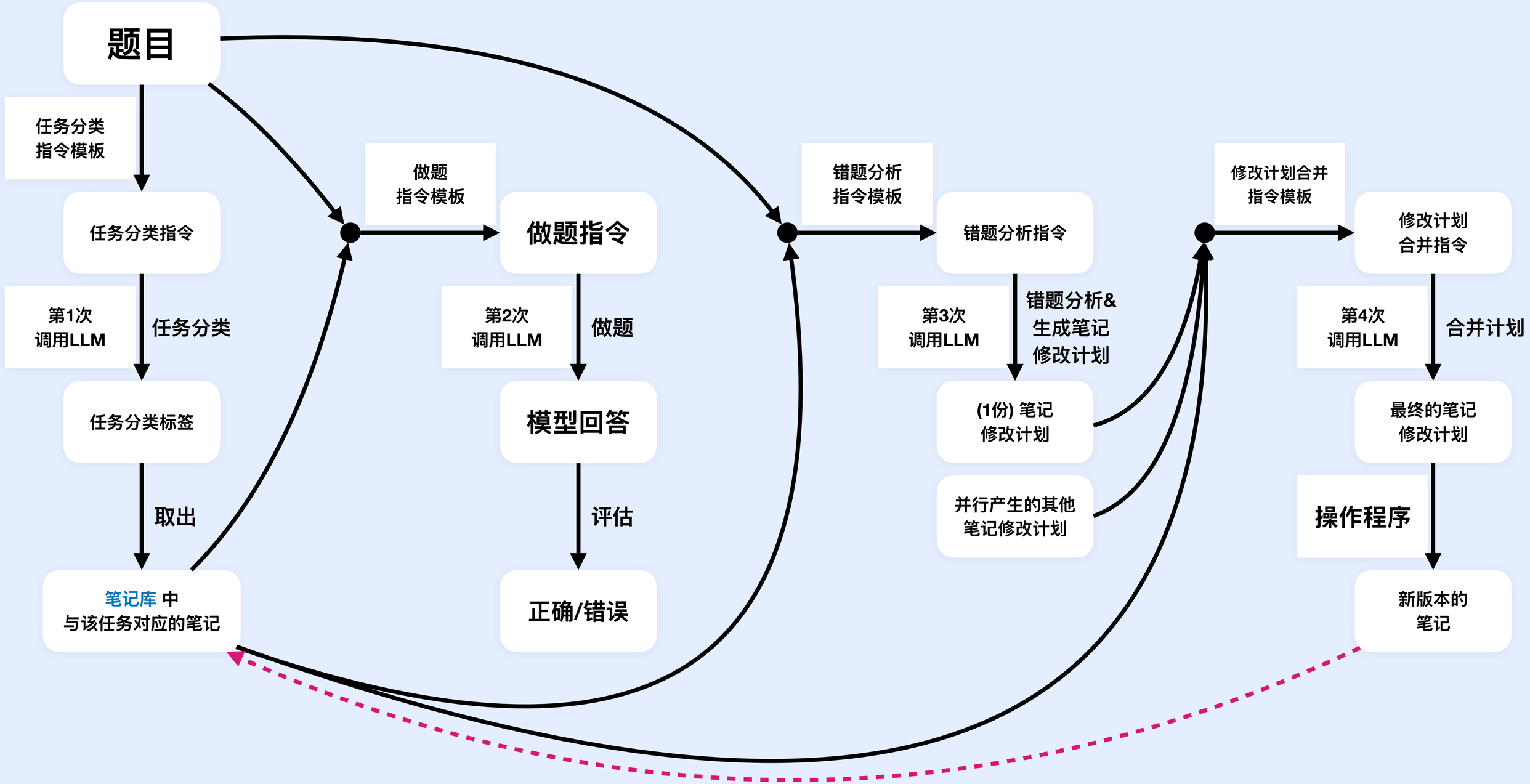
1、通过做题并分析错题，来更新一份笔记

2、可以批量处理，并行做题

3、可观察模型表现的变化，理论上可实现自动训练

SWOT 系统的工作原理

SWOT 系统（目前）的工作原理



SWOT 第1次调用LLM：任务分类

System

任务描述
判断当前题目是否属于现有题型中的某个题型（但不用做题）。

user是谁
user并不是通常意义上的人类用户，而是一个数据接口，也是你的助理，它会为你提供现有的题型体系和要判断的题目数据。

你的回应
你应该回复一个标注的JSON对象（不带其他任何内容），符合以下接口定义：

```TypeScript
interface YourResponse {
  analyze: string;  // 你的初步分析。
  matched: boolean;  // 是否找到匹配的题型
  name?: string|null|undefined;  // 匹配的题型名称
}
```

User

====[题型体系]====
(如果有笔记，则抽取每条笔记的 **name, desc, clue** 三个字段)
(如果没有笔记，则留空)
====[题目]====
(JSON格式的题目内容)
====[请按要求回应]====

Assistant

(模型的回复)

SWOT 第2次调用LLM：做题

System

任务描述
你需要结合现有的笔记，完成题目，并按要求回复。
- 你应该遵循笔记中的步骤描述（steps）来完成题目，这些步骤是伪代码，而你可以选择使用自然语言、结合题目内容来阐述这些步骤。
- 你可以假定存在笔记中定义的工具函数（tools），通过扮演和模拟来使用它们。
- 你需要检查笔记中的数据记录（datums），看看是否用得上。
- 你要注意笔记中提到的提示（tips），并在答题时遵循它们。
- 笔记不一定完全可靠，你需要灵活变通。
- 如果没有匹配的题型，或者笔记不够清晰，则需要自己思考，建立答题思路。

user是谁
user并不是通常意义上的人类用户，而是一个数据接口，也是你的助理，
它会为你提供现有的笔记和要完成的题目。

你的回应
你应该回复一个标注的JSON对象（不带其他任何内容），符合以下接口定义：
\`\`\`TypeScript
interface YourResponse {
  plan: string; // 你的初步计划，描述你打算如何完成这个题目。简要描述即可。
  usefulDatums?: string[]; // 如果笔记中有某些datum对完成这个题目有帮助，你可以在这里用自然语言描述。
  analyzes: string[]; // 详细的解题过程，应该遵循笔记中的步骤，需要结合题目具体内容详细具体地阐述。
  answer: string; // 你的最终答案，通常是一个字符串。
  didFollow: boolean; // 是否遵循了笔记的思路。
  reflection?: string; // 【非必要】你对这个题目的反思，描述你在完成这个题目时的思考过程和收获。
  noteAdvice?: string; // 【非必要】你对笔记的建议，描述你认为笔记中哪些地方需要改进。
}

User

====[笔记]====
(上一步骤所确定题型对应的完整笔记)
====[题目]====
(JSON格式的题目内容)
====[请按要求回应]====

Assistant

(模型的回复)

SWOT 第3次调用LLM：分析错题，生成笔记修改计划

System

任务描述
总任务：根据错题和正确答案，作充分的分析，并规划笔记的修改方案。按要求回复。
如果要修改笔记，你需要使用特定的笔记操作算子（见下文）来完成修改，并以JSON对象的形式嵌入在你的回应中。
笔记修改的重点应该是对题型的区分，以及围绕解题步骤的完善。
关于笔记
${NOTE_DESC_TOKEN}
笔记操作
${NOTE_OPS_TOKEN}
user是谁
user并不是通常意义上的人类用户，而是一个数据接口，也是你的助理，
它会为你提供题目、正确答案（可能带有解释说明）、错误答案和现有的笔记。
你的回应
你应该回复一个标注的JSON对象（不带其他任何内容），符合以下接口定义：
\`\`\`TypeScript
interface YourResponse {
  glance: string; // 简要描述你对错误情况的第一印象。
  format_analyze: string; // 检查正确答案和错误答案，评估是否仅是由于格式问题导致的错误。如果存在这种错误，则应该完善题型的name、desc或clue（判别依据）。
  analyze0: string; // 详细讨论为什么现有的笔记不足以解决这个题目。针对steps, tools, datums和tips逐个进行分析。
  // (此处省略若干行)
  analyze_final: string; // 你的总结。
  operations?: {
    method: string; // 如 "CREATE_QT" 等，是你计划对笔记执行的操作。
    args: {[key: string]: any}; // 你计划对笔记执行的操作的参数，参见 笔记操作 一节。
  }[]; // 你计划对笔记执行的若干操作。如果你认为没必要修改，则可以不返回这个字段。
}

User

====[笔记]====
(第1个步骤所确定题型对应的完整笔记)
====[题目、正误答案、解析]====
(JSON格式的题目内容、标准答案、错误答案、标准答案的解释)
====[请按要求回应]====

Assistant

(模型的回复)

SWOT 第4次调用LLM：多个笔记修改方案的合并

System

任务描述

总任务：结合原始笔记，把不同的笔记修改计划合并且完善成单独一份笔记修改计划。\n0、你将看到若干名做题专家对于一份答题笔记的修改计划。专家们的工作是并行的，所以可能存在重复。\n1、你要评估来自不同专家的笔记修改计划之间的共性和差异。\n2、尤其注意如果有专家要创建新的题型，要看看其他专家是否也想创建类似的题型，但使用了不同的题型名称。\n3、你需要把所有专家的修改计划合并成一份新的笔记修改计划，覆盖他们所提到的所有修改，但去除了实际含义有重复的部分。\n4、你还要考虑原始笔记的情况，以及不同修改方案和原始笔记之间的叙述风格。\n5、如果笔记的格式或风格混乱或不一致，你需要自己额外制定修改计划进行整理。

关于笔记\n${NOTE_DESC_TOKEN}

笔记操作\n${NOTE_OPS_TOKEN}

user是谁\nuser并不是通常意义上的人类用户，而是一个数据接口，也是你的助理，\n它会为你提供现有的笔记和专家们的修改计划（并行）。

完善笔记所应考虑的方面

- (此处省略若干行)

你的回应

你应该回复一个标注的JSON对象（不带其他任何内容），符合以下接口定义：

\`\`\`TypeScript
interface YourResponse {
  /** 你对不同专家修改计划的分析和大致的合并思路。 */
  analyze1: string;
  /** 你根据原有笔记内容及风格所做的额外分析及进一步完善的思路 */
  analyze2: string;
  /** 经过你合并和完善之后的修改计划。 */
  operations?: {
    method: string; // 如 "CREATE_QT" 等，是计划对笔记执行的操作。
    args: {[key: string]: any}; // 计划对笔记执行的操作的参数，参见 笔记操作 一节。
  }[];
}

User

====[笔记]====

(第1个步骤所确定题型对应的**修改之前的**完整笔记)

====[并行修改计划]====

(JSON格式，多个笔记修改方案的列表)

====[请按要求回应]====

Assistant

(模型的回复)

SWOT 系统的工作原理

效果影响因素：

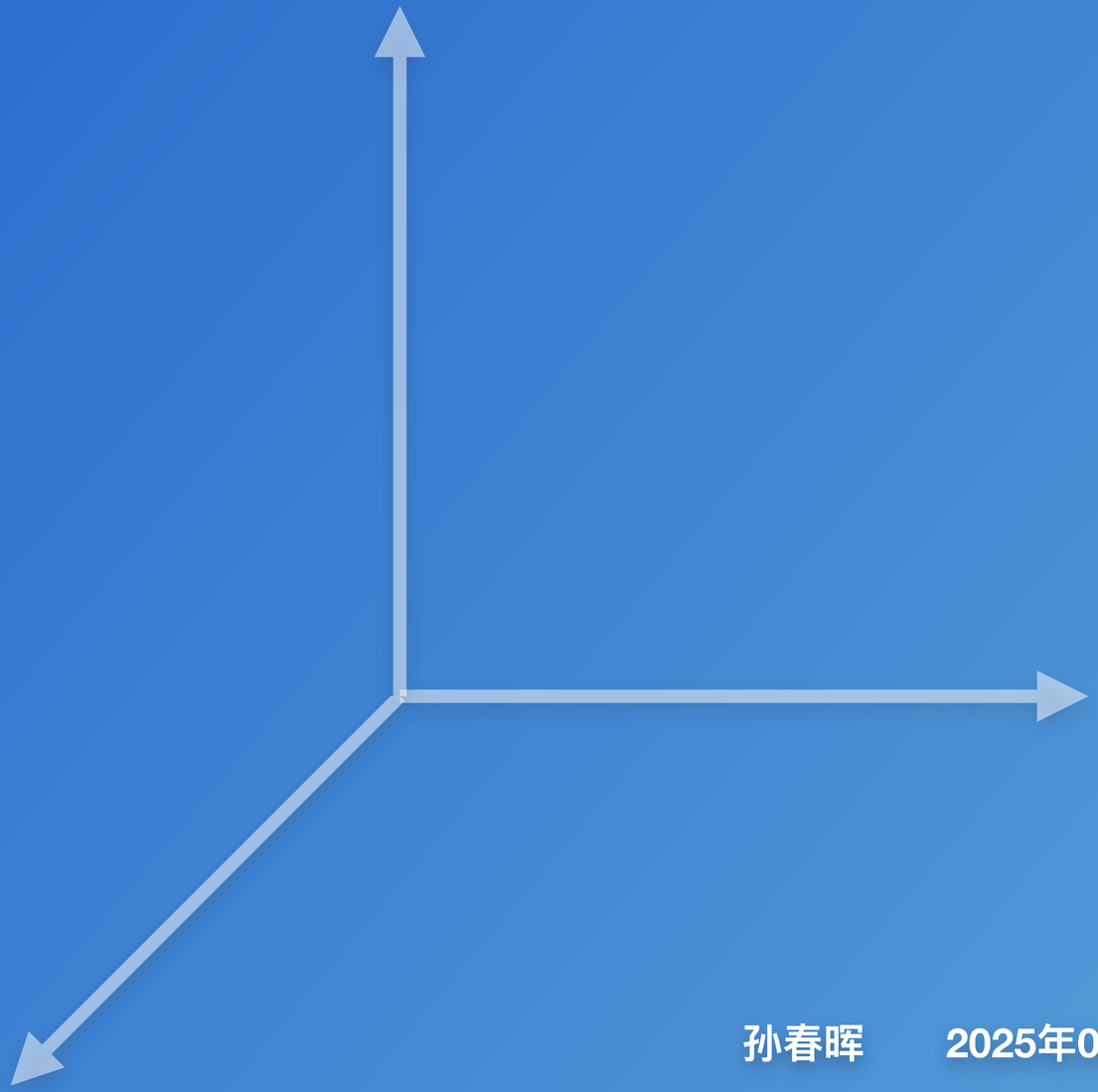
1、系统提示词

2、上下文整体结构

3、是否带有答题历史

4、JSON格式指令

5、模型调用时的参数



谢谢！

请大家批评指正

孙春晖　　2025年05月29日　　北京大学　现代汉语 (计算语言学 | 中文信息处理) 讨论班