

关于大模型系列

- ・内容背景
 - LLM + Al Agent: 大模型遇到智能体
- ・具体内容
 - | Al Agent 组成介绍:LLM + 记忆 + 规划 + 工具
 - 2. Al Agent 规划手段: Task Decomposition 与 Self Reflection
 - 3. Al Agent 热门应用:交互式 Agent、自动化 Agent 与多模态 Agent
 - 4. Al Agent 问题与挑战: Agent 的问题、 Agent 的局限性

ZOMI

大模型业务全流程

3. 模型算法

多模态

一切皆Tokens

LLM模型

架构

数据 & 模型算法 模型训练 & 微调 模型验证 & 推理部署 2. 数据处理 4. 模型训练 6. 模型微调 7. 模型验证 开源 数据 向量 下游 测评 梯度 梯度 混合 全参微调 数据 预处理 数据库 任务 标准 精度 检查 累积

1. AI 集群建设

计算、存储、 网络

AI 集群机房建 设

AI 集群上线与 运维

5. 分布式并行

训练集群稳定性

集群算力准备



低参微调

指令微调

8. 推理与智能体

推理

加速

量化

压缩

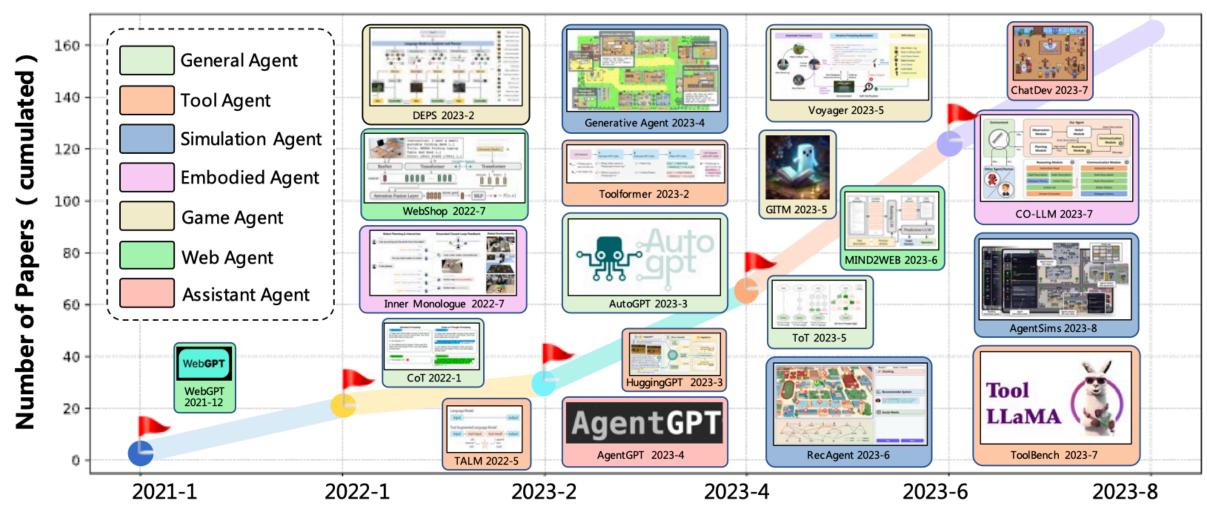
Course chenzomi 12.github.io

bench

mark

9. Agent

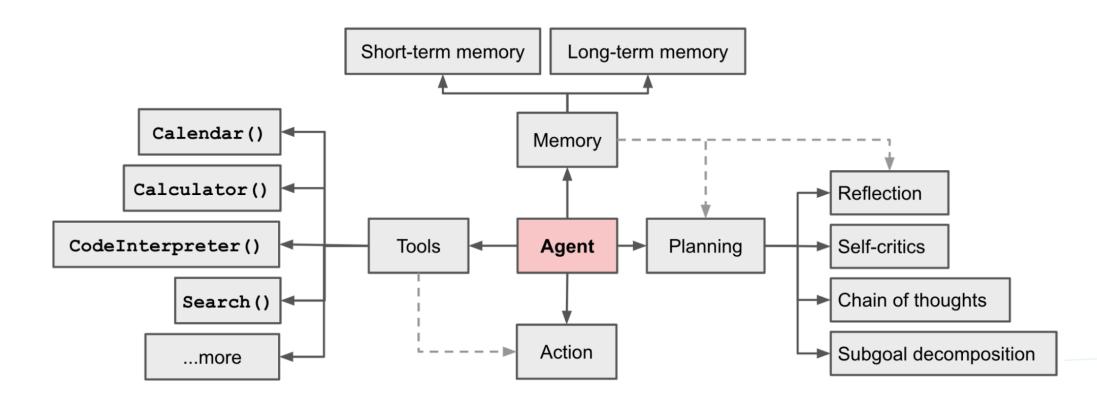
智能体



Time (Year-Month)

关键组成

• 规划 Planning + 记忆 Memory + 工具 Tools



规划 Planning

- · 规划:一项复杂任务通常包括多个子步骤, Agent 需要提前将一项任务分解为多个子任务。
 - · **子目标与分解(Subgoal and decomposition)**: Agent 将复杂任务分解为更小、更易于处理的子目标,从而实现对复杂任务的高效处理。
 - **反思与完善(Reflection and refinement)**: Agent 可以对历史的动作进行自我批评和自我反思,从错误中吸取教训,并为未来的步骤进行改进,从而提高最终结果的质量。

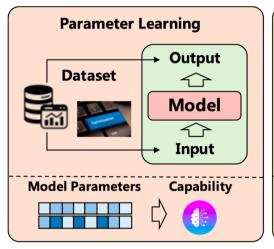
· 实现:通过prompt engine来引导 LLM 实现规划(即步骤分解)。



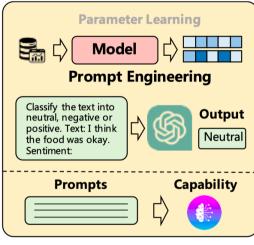
1. Agent & Prompt

Agent 智能体与 Prompt 提示词的关系

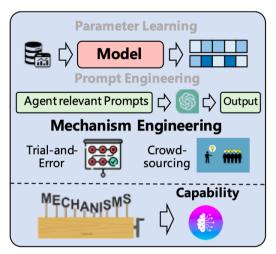
- Al Agent 绝大多数创新点还是在 Prompt 层面,即:通过更好的 Prompt 提示词来激发 LLM 模型的能力。通过 Prompt 提示词,让 LLM 大模型仿照 Prompt 给出的方式来执行的一种应用范式。
- Prompt 里面包含关于 Tools 的描述,最后 Al Agent 智能体就可以根据模型的输出使用外部 Tools (例如计算器,搜索API,数据库,程序接口,各种模型的API) 能使用外部 API 或者知识库。







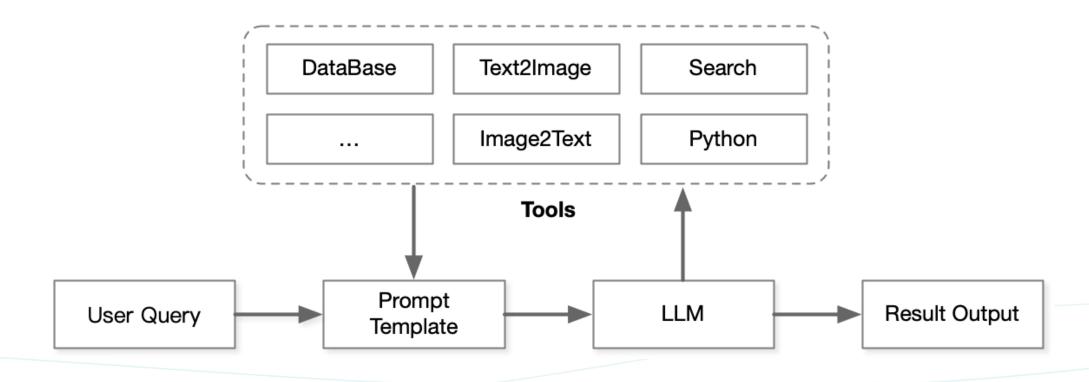
The era of large language model



The era of agent

Agent 智能体与 Prompt 提示词的关系

图中展示了一个包含 Al Agents 智能体系统,通过 Tools让 LLM 大模型获取外部输入,通过 Prompts 来驱动 LLM 大模型得到用户想要的输出。



- 一项复杂的任务通常涉及许多步骤
- · Agent 必须了解任务是什么并提前进行规划



Prompt 提示词

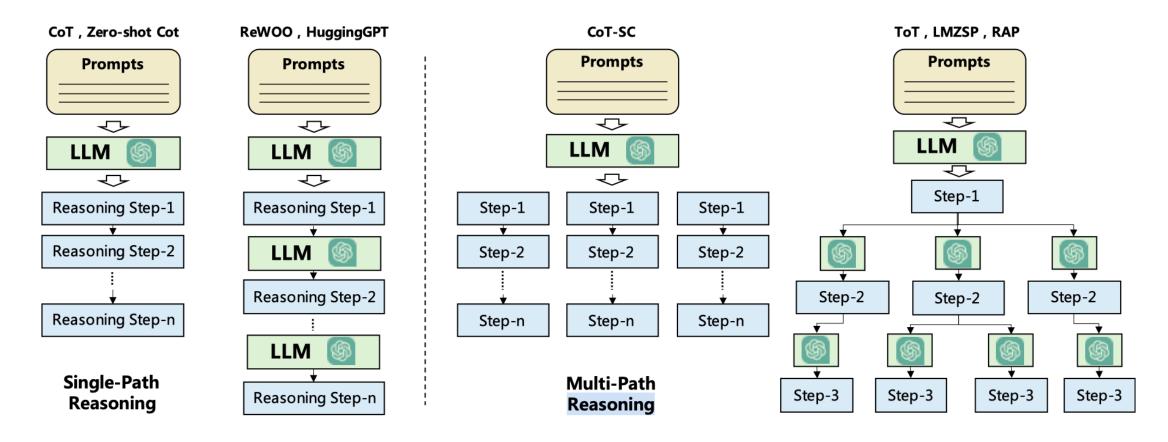


Figure 3: Comparison between the strategies of single-path and multi-path reasoning. LMZSP represents the model proposed in [70].

2. 规划 Planning

一任务分解

任务分解1:思维链 CoT

- **思维链(Chain-of-thought,CoT)一**种改进的提示策略,用于提高 LLM 在复杂推理任务中的性能,如算术、常识和符号推理。**CoT 已经成为增强复杂任务上模型性能的标准提示技术。**
- 实现过程中,模型被指示「一步一步思考」,从而将困难任务分解为更小、更简单的步骤。在
 Al Agent 中 CoT 将大型任务转化为多个可管理的小任务,并解释清楚模型的思维过程。

任务分解1:思维链 CoT

相比于上下文学习,思维链多了中间的中间的推导提示:

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27.



Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

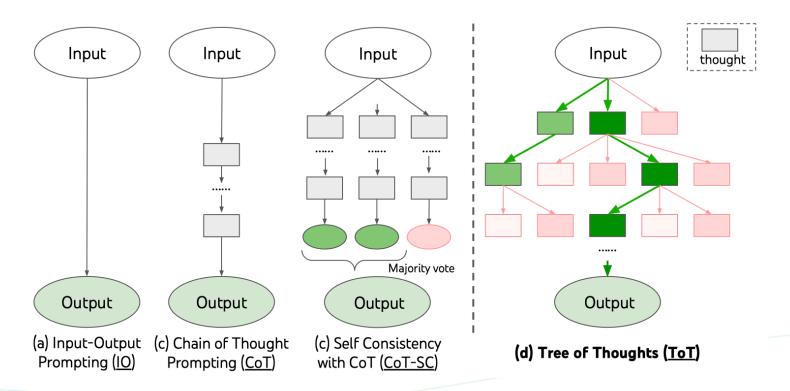
A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. 🗸

任务分解II:思维树 ToT

- 思维树(Tree of Thoughts, ToT),通过同时考虑多个可能的 Plan,并利用价值反馈机制进行决策,扩展了现有的规划方法。即通过每一步探索多种推理可能性来扩展 CoT。
- 首先将问题分解为多个思考步骤,并在每个步骤中生成多个思考,创建一种树结构。搜索过程可以是广度优先搜索(BFS)或深度优先搜索(DFS),其中每个状态由分类器或多数 vote 进行评估。具体任务分解过程可以通过以下三种方式完成:
 - I. 基于 LLM 提示,如「XXX的步骤是什么?」、「实现 XXX的子目标是什么?」
 - 2. 使用特定于任务的提示,比如「写一个XXX故事大纲」
 - 3. 人工输入。

任务分解II:思维树 ToT

思维树(Tree of Thoughts, ToT),通过同时考虑多个潜在的可行计划,并利用价值反馈机制进行决策,扩展了现有的规划方法。即通过每一步探索多种推理可能性来扩展 CoT。



Tree of Thoughts: Deliberate Problem Solving with Large Language Models

3. 规划 Planning

自我反省

自我反思 Self Reflection

- 自我反思 (Self-reflection) 允许 Agent 通过完善以往行动决策和纠正以往错误来迭代改进。
 - ReAct:通过将动作空间扩展为一个任务特定的「离散动作和语言空间的组合」;
 - 2. Self-ask:是一种follow-up的使用范式,仅仅包含follow-up, immediate answer步骤;

自我反思 I: ReAct

- ReAct:将动作空间扩展为任务特定的「离散动作和语言空间的组合」,将推理和动作集成在
 LLM中。离散动作使 LLM 能够与环境交互,而语言空间促使 LLM 以自然语言生成推理轨迹。
- Thought: ... Action: ... Observation: (Repeated many times)

```
(1d) ReAct (Reason + Act)
Thought 1: I need to search Apple Remote and find the
program it was originally designed to interact with.
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in
October 2005 by Apple ... originally designed to control the
Front Row media center program ...
Thought 2: Apple Remote was originally designed to control
the Front Row media center program. I need to search
Row next and find what other device can control it.
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row
Seat to Earth', 'Front Row Motorsports',' Front Row
(software) ', ...]
Thought 3: Front Row is not found. I need to search Front
Row (software) .
Act 3: Search[Front Row (software)
Obs 3: Front Row is a discontinued media center software ...
Thought 4: Front Row (software) is controlled by an Apple
Remote or the keyboard function keys. So the answer is
keyboard function keys.
Act 4: Finish[keyboard function keys]
```

```
(2b) ReAct (Reason + Act)
Act 1: Think[First I need to find a pepper shaker...more
likely to appear in cabinets (1-6), countertops (1-3), ... ]
Act 2: Go to cabinet 1
Obs 2: On the cabinet 1, you see a vase 2.
(Here the agent go to cabinet 1, then cabinet 2, 3, then countertop 1 and 2)
Act 6: Go to countertop 3
Obs 6: On the countertop 3, you see a apple 1, a bread 1,
..... a pepper shaker 1, and a vase 3.
Act 7: Take pepper shaker 1 from countertop 3
Obs 7: You pick up the pepper shaker 1 from the countertop 3.
Act 8: Think [Now I find a pepper shaker 1. Next, I need to
put it in/on drawer 1. ]
Act 9: Go to drawer 1
Obs 9: Drawer 1 is closed.
Act 10: Open drawer 1
Obs 10: You open Drawer 1 ...
Act 11: Put pepper shaker 1 in/on drawer 1
Obs 11: You put pepper shaker 1 in/on the drawer 1.
```

自我反思 I: ReAct

• Thought: ... Action: ... Observation: (Repeated many times)

知识密集型任务(HotpotQA、FEVER)

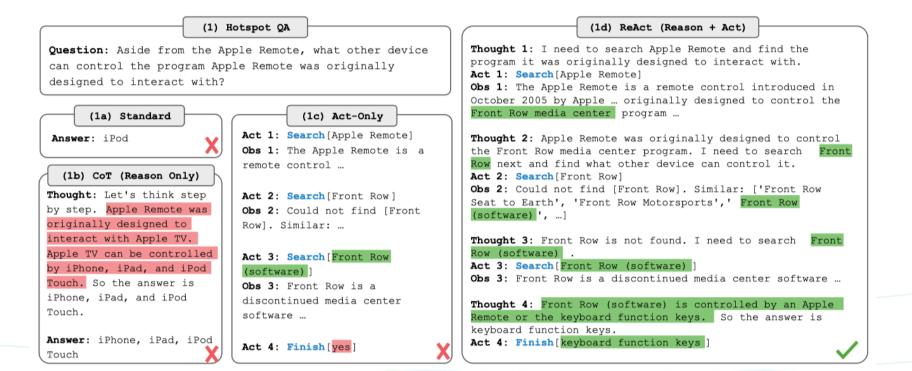
(1d) ReAct (Reason + Act) Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with. Act 1: Search [Apple Remote] Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the Front Row media center program ... Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Row next and find what other device can control it. Act 2: Search[Front Row] Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports',' Front Row (software) ', ...] Thought 3: Front Row is not found. I need to search Front Row (software) . Act 3: Search[Front Row (software) Obs 3: Front Row is a discontinued media center software ... Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys. Act 4: Finish [keyboard function keys]

决策型任务(AlfWorld Env、WebShop)

```
(2b) ReAct (Reason + Act)
Act 1: Think[First I need to find a pepper shaker...more
likely to appear in cabinets (1-6), countertops (1-3), ... ]
Act 2: Go to cabinet 1
Obs 2: On the cabinet 1, you see a vase 2.
(Here the agent go to cabinet 1, then cabinet 2, 3, then countertop 1 and 2)
Act 6: Go to countertop 3
Obs 6: On the countertop 3, you see a apple 1, a bread 1,
..... a pepper shaker 1, and a vase 3.
Act 7: Take pepper shaker 1 from countertop 3
Obs 7: You pick up the pepper shaker 1 from the countertop 3.
Act 8: Think [Now I find a pepper shaker 1. Next, I need to
put it in/on drawer 1. ]
Act 9: Go to drawer 1
Obs 9: Drawer 1 is closed.
Act 10: Open drawer 1
Obs 10: You open Drawer 1 ...
Act 11: Put pepper shaker 1 in/on drawer 1
Obs 11: You put pepper shaker 1 in/on the drawer 1.
```

自我反思 I: ReAct

- ReAct本质融合 Reasoning 和 Acting 的一种 Prompt 范式,推理过程是浅显易懂,仅仅包含thoug ht-action-observation步骤,很容易判断推理的过程的正确性。四种 Prompt 比较:
 - (I) Native Prompt; (2) Chain-of-thought; (3) Acting Only; (4) ReAct。



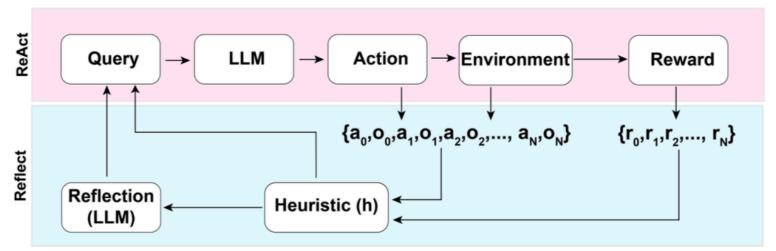


自我反思 I: ReAct vs CoT

- Chain-of-Thought 推理只是作为静态黑盒,它没有用外部知识,所以在推理过程中会出现事实幻想(fact hallucination)和错误传递(error propagation)的问题。
- ReAct 克服了在思维链推理中的幻觉和错误传播问题,通过与简单的维基百科API交互,生成类似于人的任务解决型轨迹,解释性进一步增强。

自我反思 I: ReAct 的扩展 Reflexion 框架

• Reflexion 框架则为 Agent 配备了动态记忆和自我反思能力,提高 Reasoning 能力。通过借鉴 RL 流程,奖励模型提供简单的二元奖励,特定于任务的动作空间通过 LLM 进行增强,实现复杂推理步骤,动作空间遵循 ReAct 中的配置(Thought... Action ... Observation);在每个动作 a_t 之后,智能体计算启发式 h_t,并选择性地根据自我反思结果来决定重置环境,从而开始新的试验。



Reflexion: Language Agents with Verbal Reinforcement Learning

自我反思 II: Self-ask

• Self-ask是一种follow-up的使用范式,仅仅包含 follow-up, immediate answer步骤,至于 follow-up 多少个 step , 完全由 Agent 自己决定。

Direct Prompting

Question: Who lived longer, Theodor Haecker or Harry Vaughan Watkins?
Answer: Harry Vaughan Watkins.

Question: Who was president of the U.S. when superconductivity was discovered?
Answer: Franklin D. Roosevelt

Chain of Thought

Question: Who lived longer, Theodor Haecker or Harry Vaughan Watkins?

Answer: Theodor Haecker was 65 years old when he died. Harry Vaughan Watkins was 69 years old when he died.

So the final answer (the name of the person) is: Harry Vaughan Watkins.

Question: Who was president of the U.S. when superconductivity was discovered?

Answer: Superconductivity was discovered in 1911 by Heike Kamerlingh Onnes. Woodrow Wilson was president of the United States from 1913 to 1921. So the final answer (the name of the president) is: Woodrow Wilson.

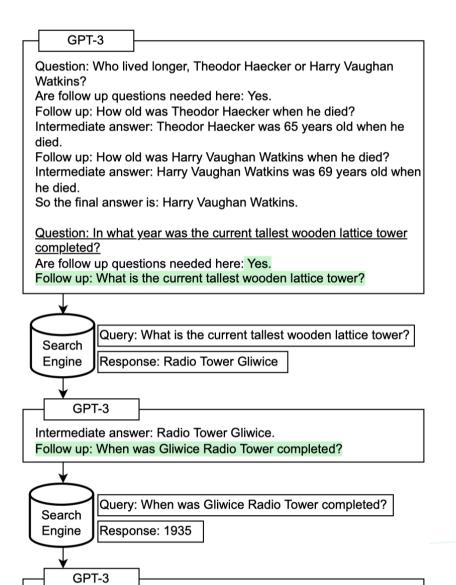
Self-Ask

GPT-3 Question: Who lived longer, Theodor Haecker or Harry Vaughan Watkins? Are follow up questions needed here: Yes. Follow up: How old was Theodor Haecker when he died? Intermediate answer: Theodor Haecker was 65 years old when he died. Follow up: How old was Harry Vaughan Watkins when he died? Intermediate answer: Harry Vaughan Watkins was 69 years old when he died. So the final answer is: Harry Vaughan Watkins Question: Who was president of the U.S. when superconductivity was discovered? Are follow up questions needed here: Yes. Follow up: When was superconductivity discovered? Intermediate answer: Superconductivity was discovered in 1911. Follow up: Who was president of the U.S. in 1911? Intermediate answer: William Howard Taft. So the final answer is: William Howard Taft.

白色背景为 prompt , 绿色背景为 LLM 输出 , 下划线为 inference-time ; Self-ask 需要一个/少量 Prompt 来引导 LLM 如何回答 Prompt 问题。

自我反思 II: Self-ask

- 前面的是 inference-time question, Prompt 末 尾插入短语 "Are follow up questions needed h ere:", 可以略微改善结果。
- LLM 输出一个 Response: Yes, 这意味着后续 行动问题是必要的。
- 然后 LLM 输出第一个 follow-up 问题,使用搜索引擎找到答案,作为 Prompt 继续 follow-up 问题,直到它决定有足够的信息为止。
- 最终引导 LLM 输出: "So the final answer is:", 使得最终答案可以很容易根据 Self ask 解析出来。



Intermediate answer: 1935. So the final answer is: 1935.



把AI系统带入每个开发者、每个家庭、 每个组织,构建万物互联的智能世界

Bring AI System to every person, home and organization for a fully connected, intelligent world.

Copyright © 2023 XXX Technologies Co., Ltd. All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. XXX may change the information at any time without notice.



ZOMI

Course chenzomil2.github.io

GitHub github.com/chenzomi12/DeepLearningSystem