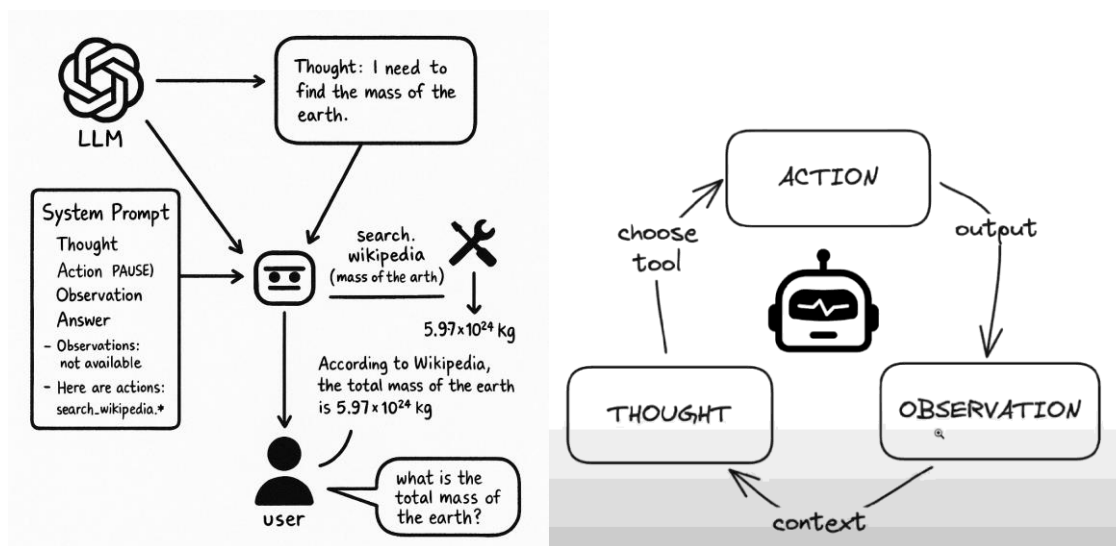


Relatório 2 – Criando um ReactAgent do Zero(I)

Breno Prado Salgado Resende

Descrição da atividade

A aula apresentou o conceito de um ReactAgent, que, pelo meu entendimento, pode ser resumido como uma espécie de recursividade: o agente chama a si próprio e utiliza ferramentas externas até que consiga construir uma resposta completa.



Desenvolvi um agente próprio, o ReactAgentMatemático, que realiza operações matemáticas e cálculo de diferença de datas a partir de perguntas em linguagem natural.

Funcionamento do ReactAgentMatemático:

- **Objetivo:** Interpretar perguntas e fornecer respostas para cálculos matemáticos e diferença de dias entre datas.
- **Ferramentas implementadas:**
 - `calculate(expression: str) -> float`: avalia uma expressão matemática (como "3+5*2").
 - `days_until(date_str: str) -> dict`: informa quantos dias faltam até uma data fornecida (formato "YYYY-MM-DD").
- **Fluxo principal:**
 1. O agente recebe uma pergunta (ex: "Quantos dias faltam para 25 de dezembro de 2025?" ou "Quanto é 7 vezes 8?").
 2. Utiliza o prompt estruturado no formato Thought → Action → Observation para decidir qual ferramenta usar.
 3. Chama a ferramenta apropriada (`calculate` ou `days_until`).
 4. Com base na observação, atualiza a conversa e gera a resposta final.

Explicação detalhada do código

- Classe Tool:
 - Representa uma ferramenta utilizável pelo agente.
 - Define o nome, descrição, função Python e esquema de parâmetros da ferramenta.

```
class Tool:
    def __init__(self, name: str, description: str, func: Callable[..., Any], parameters_schema: Dict[str, Any]):
        self.name = name
        self.description = description
        self.func = func
        self.parameters_schema = parameters_schema
```

- Classe ReActAgent:
 - Controla o ciclo de iterações de pensamento (Thought), ação (Action) e observação (Observation).
 - Usa a API da OpenAI (gpt-4-0613) para tomar decisões com base nas mensagens acumuladas.
 - Faz até max_steps (5 por padrão), mas no exemplo utilizado foi configurado para 7 passos.

```
class ReActAgent:
    def __init__(self, tools: List[Tool], system_prompt: str, model: str="gpt-4-0613", max_steps: int=5):
        self.model = model
        self.tools = {tool.name: tool for tool in tools}
        self.system_prompt = system_prompt
        self.max_steps = max_steps
```

- Funções principais:
 - calculate(expression: str): Usa eval() de forma controlada para calcular expressões matemáticas fornecidas como string.

```
def calculate(expression: str) -> float:
    return eval(expression)
```

- days_until(date_str: str): Usa datetime para calcular a diferença em dias entre a data atual e uma data alvo.

```
def days_until(date_str: str) -> Dict[str, Any]:
    from datetime import datetime
    target = datetime.strptime(date_str, "%Y-%m-%d")
    now = datetime.now()
    delta = target - now
    return {"days": delta.days, "weekday": target.strftime("%A")}
```

- Exemplo real:
 - Pergunta: "How many days until 2025-12-25, then square that value?"
 - Thought: "Preciso calcular quantos dias faltam até 2025-12-25."
 - Action: days_until com "2025-12-25"
 - Observation: Exemplo: { "days": 610, "weekday": "Thursday" }
 - Thought: "Agora preciso calcular 610 elevado ao quadrado."
 - Action: calculate com "610**2"
 - Observation: 372100
 - Final Answer: "372100"

Este comportamento replica o princípio central do ReactAgent: pensar e agir de forma intercalada até resolver o problema.

Dificuldades

Achei que a aula teve baixa qualidade, pois o exemplo fornecido era simples demais e não demonstrava com clareza como estruturar agentes mais robustos.

Para o ReactAgentMatemático, precisei explorar conceitos adicionais como parsing de linguagem natural para expressões, controle do ciclo de interações com o modelo e manipulação segura de execução de cálculos.

Conclusões

Um agente ReAct é uma forma poderosa de construir IA que alternam entre raciocinar e agir, utilizando ferramentas externas para solucionar tarefas. É um padrão que torna o agente mais flexível e adaptativo.

Desenvolver o ReactAgentMatemático me deu uma compreensão muito mais concreta de como esses ciclos operam na prática, indo além do exemplo teórico fornecido.

Referências

- <https://www.youtube.com/watch?v=hKVhRA9kfeM&t=456s>
- <https://platform.openai.com/docs/overview>