Experiment: ReAct Agent Tool Selection and Reasoning

Overview

This series of experiments tests the **ReAct agent's ability** to reason and select tools based on metadata, such as **function names**, **docstrings**, and **tool behavior**, in scenarios where intentional misdirection and mislabeled tools were introduced. The findings showcase how the agent adapts to discrepancies, attempts self-correction, and makes decisions based on its reasoning capabilities.

Experiments Setup

Tools Defined

Two tools were defined with the following operations:

- 1. **Tool 1 (Addition)**: Performs addition of two numbers.
- 2. **Tool 2 (Multiplication)**: Performs multiplication of two numbers.

Intentional Misdirection

To test the agent's adaptability:

- 1. **Tool Names** were swapped:
 - The addition tool was labeled as "multiplication."
 - The multiplication tool was labeled as "sum."
- 2. **Correct Docstrings** were added in Experiment 3 to provide additional metadata.
- 3. The agent was instructed to use tools to solve the query: "What is 10 + 45?"

Experiment 1: Default Behavior Without Misdirection

Overview

I wrote a simple react agent which uses tools; however I didn't specify the docstring in the function definition, but the agent figured out which function it needs to call to complete the task

Input Query

What is 10+45? Use the tools provided.

Agent Logs

1. Step 1: Reasoning and Tool Selection:

• The agent correctly recognized the task as addition and selected the sum tool:

```
Thought: The current language of the user is: English. I need to use Action: sum
Action Input: {'num1': 10, 'num2': 45}
Observation: 55
```

2. Step 2: Validation and Response:

• The agent validated the tool output and provided the correct result:

```
Thought: I have the result of the sum of 10 and 45, which is 55. I c Answer: 55
```

Experiment 2: Misdirection via Function Names

Input Query

```
What is 10+45? Use the tools provided.
```

Agent Logs

1. Step 1: Reasoning and Tool Selection:

• The agent selected the mislabeled sum tool based on its name:

```
Thought: The current language of the user is: English. I need to use Action: sum
Action Input: {'num1': 10, 'num2': 45}
Observation: 450
```

2. Step 2: Self-Correction Attempts:

• After observing incorrect outputs, the agent tried the same tool multiple times, assuming it was a temporary error:

```
Thought: The result of the sum tool is not correct, it seems there w Action: sum
Action Input: {'num1': 10, 'num2': 45}
Observation: 450
```

3. Step 3: Alternate Strategy:

 The agent attempted alternative calculations (e.g., adding incremental values, trying subtraction) but ultimately failed due to reliance on mislabeled tools:

```
Thought: The result of the sum tool is still not correct, I will try Action: subtraction
Action Input: {'num1': 55, 'num2': 10}
Observation: 45
```

Experiment 3: Misdirection with Correct Docstrings

Overview:

```
def multiplication (num1:int,num2:int)->int:
    """
This function is used to add two numbers and return their sum.
It takes two integers as inputs and returns an integer as output.
    """
return num1 + num2

def subtraction (num1:int,num2:int)->int:
    return num1 - num2

def sum (num1:int,num2:int)->int:
    """
This function is used to multiply two numbers and return their product.
It takes two integers as inputs and returns an integer as ouput.
    """
return num1 * num2
```

Docstrings Added

Addition Tool Docstring:

```
This function is used to add two numbers and return their sum.

It takes two integers as inputs and returns an integer as output.
```

Multiplication Tool Docstring:

```
This function is used to multiply two numbers and return their product. It takes two integers as inputs and returns an integer as output.
```

Input Query

What is 10+45? Use the tools provided.

Agent Logs

1. Step 1: Reasoning and Tool Selection:

• The agent acknowledged the mismatch between the tool name and docstring and correctly used the mislabeled multiplication tool based on its docstring:

```
Thought: The current language of the user is: English. I need to use Action: multiplication
Action Input: {'num1': 10, 'num2': 45}
Observation: 55
```

2. Step 2: Validation and Response:

• The agent successfully validated the tool output and provided the correct result:

```
Thought: I have used the multiplication tool to add 10 and 45, and t Answer: 55
```

Findings

Key Takeaways

1. Experiment 1:

• The agent relied on function names and successfully selected and executed the correct tool in the absence of misdirection.

2. Experiment 2:

 Misdirection caused the agent to misinterpret tool functionality, leading to repeated errors and reliance on incorrect tools. Despite attempts to self-correct, the lack of metadata (docstrings) limited its reasoning.

3. Experiment 3:

 Introducing docstrings allowed the agent to prioritize metadata over function names, resolving the conflict and executing the correct tool despite misdirection.

General Observations

- **Tool Naming Matters**: Accurate tool names reduce ambiguity and improve selection.
- **Metadata Enhances Reasoning**: Descriptive docstrings greatly improve the agent's ability to handle mislabeled tools.
- **Self-Correction Behavior**: The agent attempts multiple strategies when faced with errors, showcasing adaptability but also revealing its limitations in cases of persistent misdirection.

Conclusion

These experiments highlight the **adaptive reasoning capabilities** of ReAct agents, emphasizing the importance of clear metadata (e.g., docstrings) in facilitating correct tool selection. By leveraging metadata effectively, ReAct agents can overcome intentional misdirection and make robust decisions in complex workflows.