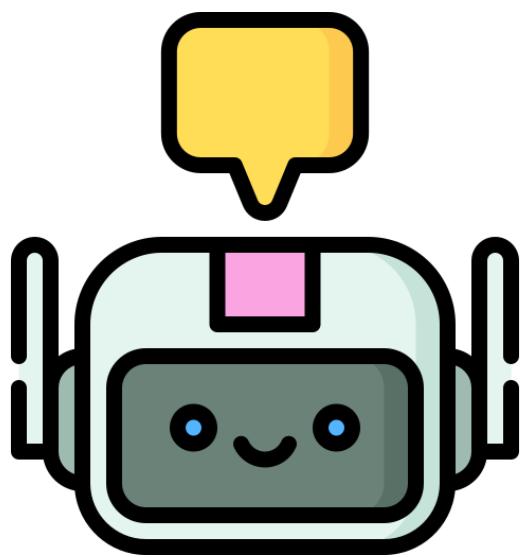


AI Agents

AI Agents Part 1 – Tutorial

Daniel



AI Agents Part - 1

AI Agents

1. What is "AI" in AI Agent?	3
2. What is "Agent" in AI Agent?	3
3. What is AI Agent?	3
4. Real-world AI agent examples	4
4.1. Self-Driving Cars (Tesla, Waymo)	4
4.2. Chatbots in Banking (HDFC's EVA, SBI's SIA)	4
4.3. Stock Trading Bots (Zerodha's Streak, Robinhood's AI Trading)	4
4.4. Healthcare AI (IBM Watson, Google DeepMind)	4
4.5. Fraud Detection in Banks (PayPal, Mastercard AI)	5
5. Key Characteristics of AI Agents	5
6. AI Agent Categories Based on Functionality	5
6.1. Reactive Agents	5
6.2. Deliberative Agents	5
6.3. Learning Agents	5
6.4. Multi-Agent Systems	5
7. The Need for AI Agents	6
7.1. Software development perspective	6
7.2. Autonomous system perspective	15
6. Core Components of AI Agents	18
6.1. Role-Playing	19
6.2. Focus	20
6.3. Tools	21
6.4. Cooperation	22
6.5. Guardrails	23
6.6. Memory	24
7. Crewai Introduction	25
8. ollama Introduction	26
9. Creating AI Agents	27
10. Environment Setup	27
11. Creating Agents in CrewAI	29
12. Single-Agent vs. Multi-Agent Workflow in CrewAI	29

AI Agents Part - 1

12.1. Single-Agent Workflow	29
12.2. Multi-Agent Workflow	29
13. AI Agents technical flow.....	30
14. Hello World Example: AI Research & Writing with CrewAI	30
14.1. AI Agents technical flow for Simple Greeting AI using CrewAI.....	30
15. Use Case 1: Automated Content Creation using CrewAI	33
15.1. AI Agents technical flow for Automated content creation using CrewAI	33
16. Use Case 2: Automated Healthcare AI Summary with CrewAI.....	41
16.1. AI Agents technical flow for healthcare content creation using CrewAI.....	41
17. Use Case 3: Multi-Agent Research Assistant with CrewAI.....	47
17.1. AI Agents technical flow for Multi-Agent Research Assistant with CrewAI.....	47

AI Agents

1. What is "AI" in AI Agent?

- ✓ AI refers to machines or software that can perform tasks that typically require human intelligence.
- ✓ These tasks include learning, reasoning, problem-solving, understanding language, and recognizing patterns.

2. What is "Agent" in AI Agent?

- ✓ An agent is something that can sense its environment, make decisions, and take actions to achieve a goal.
- ✓ An AI agent is a system that perceives the world (through sensors, data, or inputs), processes that information, and takes actions accordingly.

3. What is AI Agent?

- ✓ An AI agent is a smart system that interacts with its surroundings, learns from experience, and makes decisions to complete tasks efficiently.
- ✓ They can be software-based (like virtual assistants) or embodied in hardware (like robots).

AI Agent = AI + Agent

4. Real-world AI agent examples

- ✓ AI agents are widely used across industries to improve efficiency and automate tasks. Some key examples included here.

4.1. Self-Driving Cars (Tesla, Waymo)

- ✓ **AI Role:** Detects roads, traffic signals, pedestrians, and other vehicles using sensors and cameras.
- ✓ **Agent Behavior:** Makes driving decisions like stopping, turning, or changing lanes safely.

4.2. Chatbots in Banking (HDFC's EVA, SBI's SIA)

- ✓ **AI Role:** Understands customer queries about balances, transactions, and loans.
- ✓ **Agent Behavior:** Responds with relevant answers or directs users to human support.

4.3. Stock Trading Bots (Zerodha's Streak, Robinhood's AI Trading)

- ✓ **AI Role:** Analyses stock market trends, news, and data.
- ✓ **Agent Behavior:** Buys or sells stocks based on predicted profitability.

4.4. Healthcare AI (IBM Watson, Google DeepMind)

- ✓ **AI Role:** Reads medical reports, diagnoses diseases, and suggests treatments.
- ✓ **Agent Behavior:** Assists doctors in decision-making for better patient care.

4.5. Fraud Detection in Banks (PayPal, Mastercard AI)

- ✓ **AI Role:** Scans millions of transactions in real-time to find fraud patterns.
- ✓ **Agent Behavior:** Flags suspicious activity and prevents fraudulent transactions.

5. Key Characteristics of AI Agents

- ✓ **Autonomy** : Operate independently.
- ✓ **Perception** : Collect data from inputs.
- ✓ **Decision-making** : Analyze and act.
- ✓ **Action** : Execute tasks like responding or controlling systems.
- ✓ **Adaptability** : Improve over time.

6. AI Agent Categories Based on Functionality

6.1. Reactive Agents

- ✓ Respond to inputs without memory (e.g., rule-based chatbots).

6.2. Deliberative Agents

- ✓ Use reasoning and planning to make decisions (e.g., self-driving car AI).

6.3. Learning Agents

- ✓ Improve through experience (e.g., recommendation systems).

6.4. Multi-Agent Systems

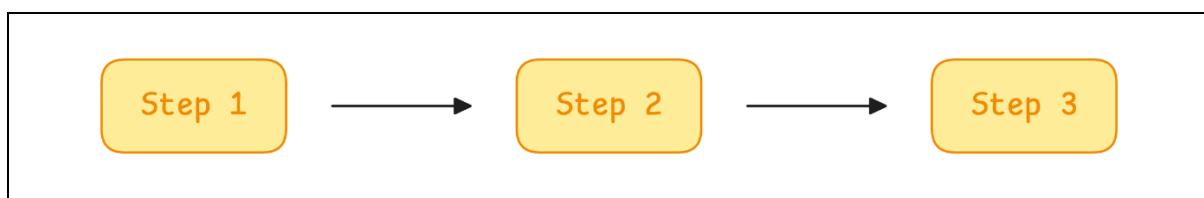
- ✓ Multiple AI agents working together (e.g., swarm robotics).

7. The Need for AI Agents

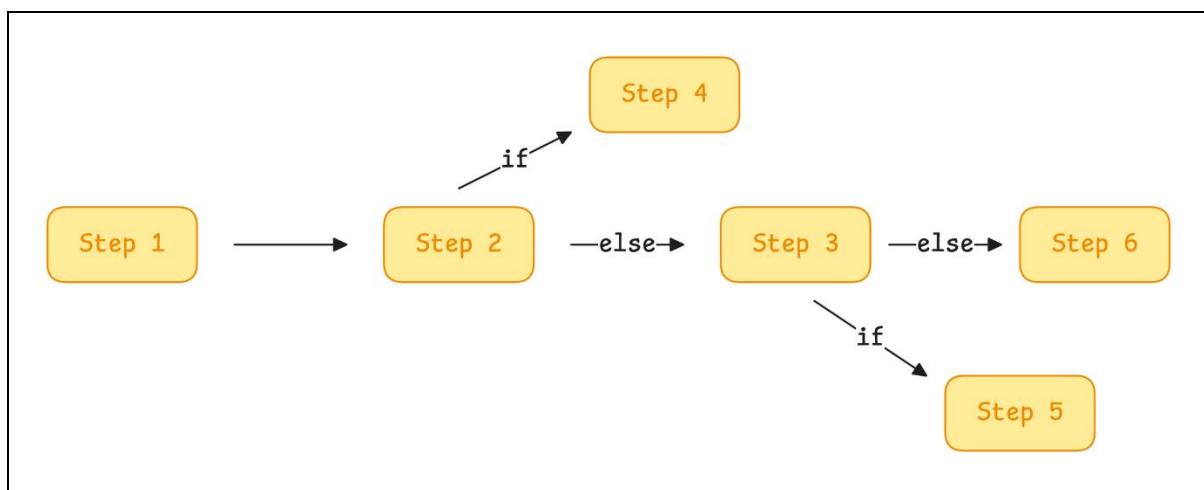
- ✓ There are several key perspectives to understanding the motivation behind AI agents:
 1. Software Development Perspective
 2. Autonomous Perspective

7.1. Software development perspective

- ✓ Think about traditional software applications, they operate based on strict, predefined rules.
 - If a program needs to complete a task, every step must be explicitly defined.



- As new scenarios arise, developers constantly need to add more conditions and logic.



- Over time, the system becomes more difficult to scale and maintain.

Traditional automation

- ✓ In other words, traditional automation follows strict predefined logic:
 - If condition A is met, do X.
 - If condition B is met, do Y.
 - Otherwise, do Z.
- ✓ Inputs have a predefined data type (text, number, etc.).
- ✓ Transformations on the input are mostly fixed.
- ✓ Output types are also fixed.



When to Stick with Traditional Automation?

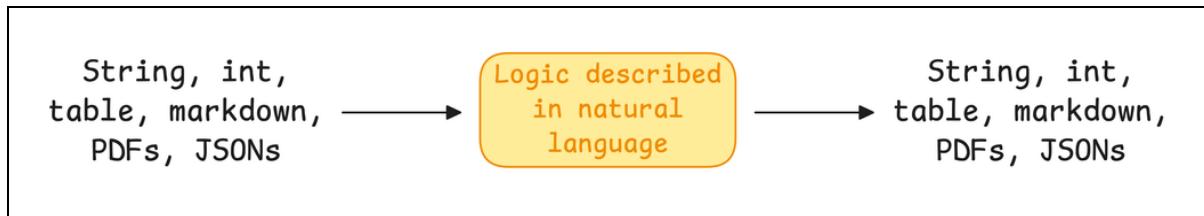
- ✓ This isn't a criticism of traditional automation, and we're not discouraging it.
- ✓ If it works for your needs, stick with it.
- ✓ There's no need to build AI agents unless truly necessary.

Why AI Agents Are Different?

- ✓ No need for explicit instructions.
- ✓ Gather information dynamically.
- ✓ Use reasoning for ambiguous problems.
- ✓ Collaborate with other agents to complete tasks.
- ✓ Leverage external tools for real-time decisions.

How AI Agents Offer More Flexibility?

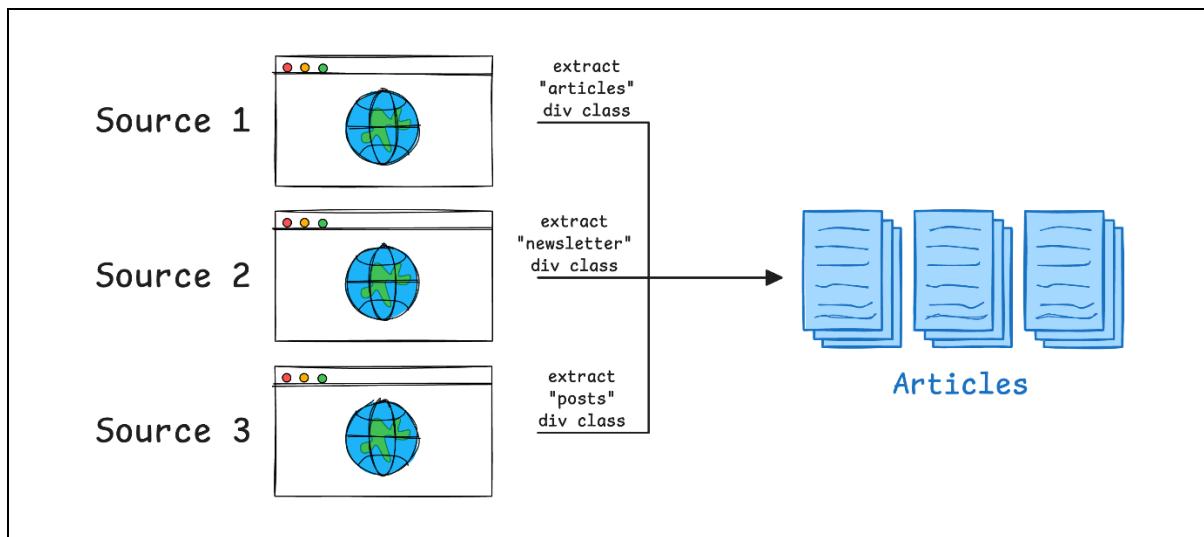
- ✓ Inputs are not limited to a specific data type.
- ✓ They can be text, numbers, PDFs, tables, markdown, JSON, and more.



- ✓ Transformations are flexible and depend on what you ask the LLM to do.
- ✓ Outputs can vary, tokens, lists, structured data, JSON, code, and more.

Traditional Approach: Manual Effort Required

- ✓ Imagine managing a news aggregation platform that collects articles from various sources.
- ✓ Traditionally, we need to manually extract and process data from different sources



- ✓ Write **scripts** to scrape multiple websites.
- ✓ Filter and categorize articles using hardcoded rules.
- ✓ Manually verify whether an article is relevant.

Limitation

- ✓ If a website changes its layout, your scraper breaks.
- ✓ If new topics arise, you must update the filtering logic manually.

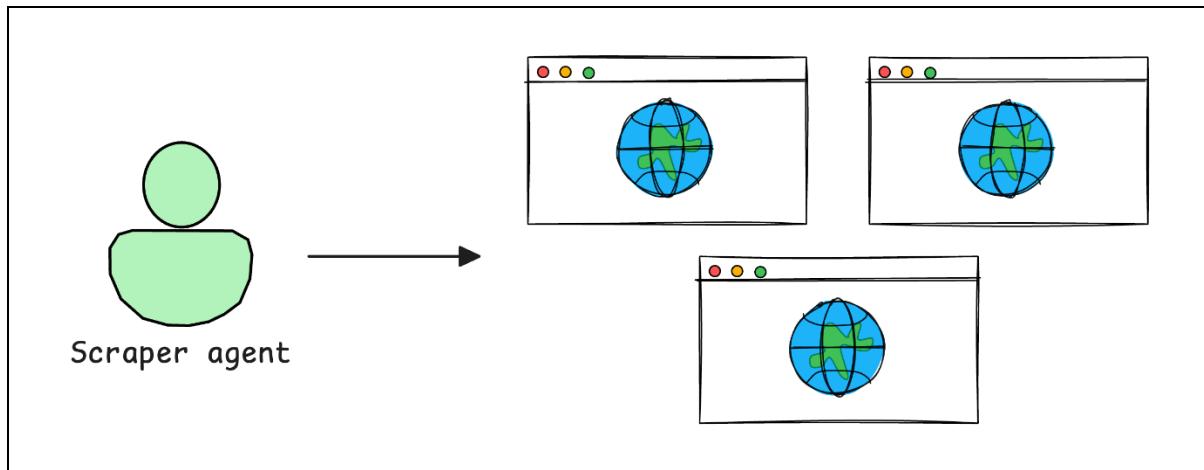
AI Agent Approach: Automation & Intelligence

How AI Agents Work Together?

- ✓ AI agents collaborate to automate the entire news processing workflow from data collection to verification and summarization.
- ✓ Each agent plays a specific role in ensuring accuracy, relevance, and efficiency:
 1. A Web Scraper Agent
 2. A Topic Analysis Agent
 3. A Fact-Checking Agent
 4. A Summarization Agent

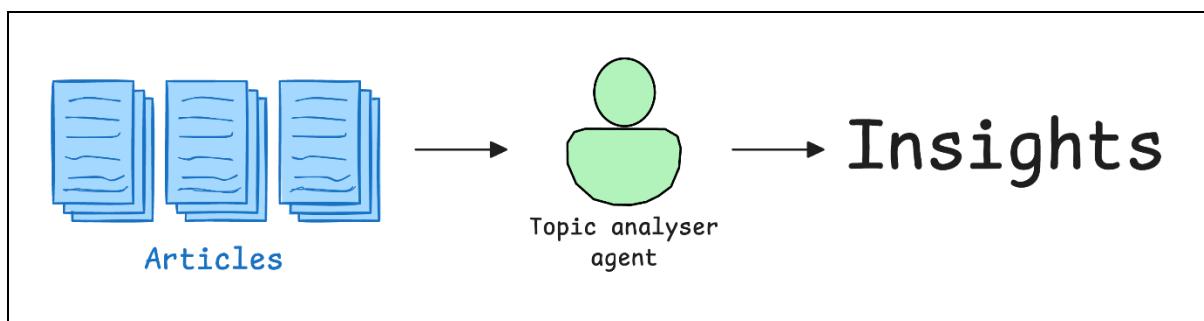
1. A Web Scraper Agent

- ✓ Automatically discovers new sources.
- ✓ Adapts to changes in web page structures without manual intervention.



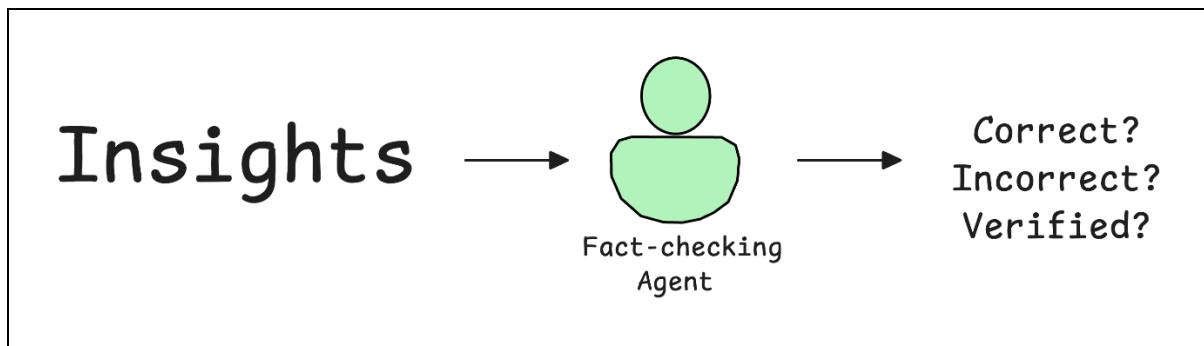
2. A Topic Analysis Agent

- ✓ Analyzes articles in real time.
- ✓ Detects emerging trends.
- ✓ Classifies content efficiently.



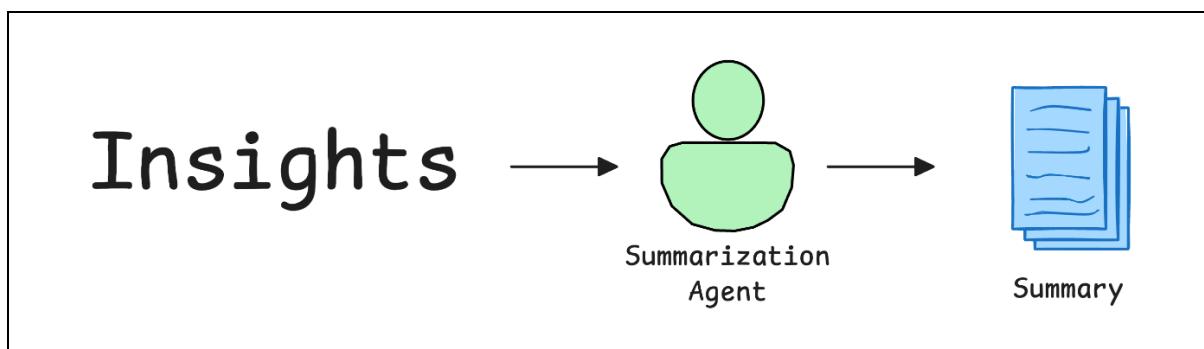
3. A Fact-Checking Agent

- ✓ It ensures article credibility by:
 - Cross-referencing external data sources.
 - Verifying claims for accuracy.
 - Detecting misinformation before publication.



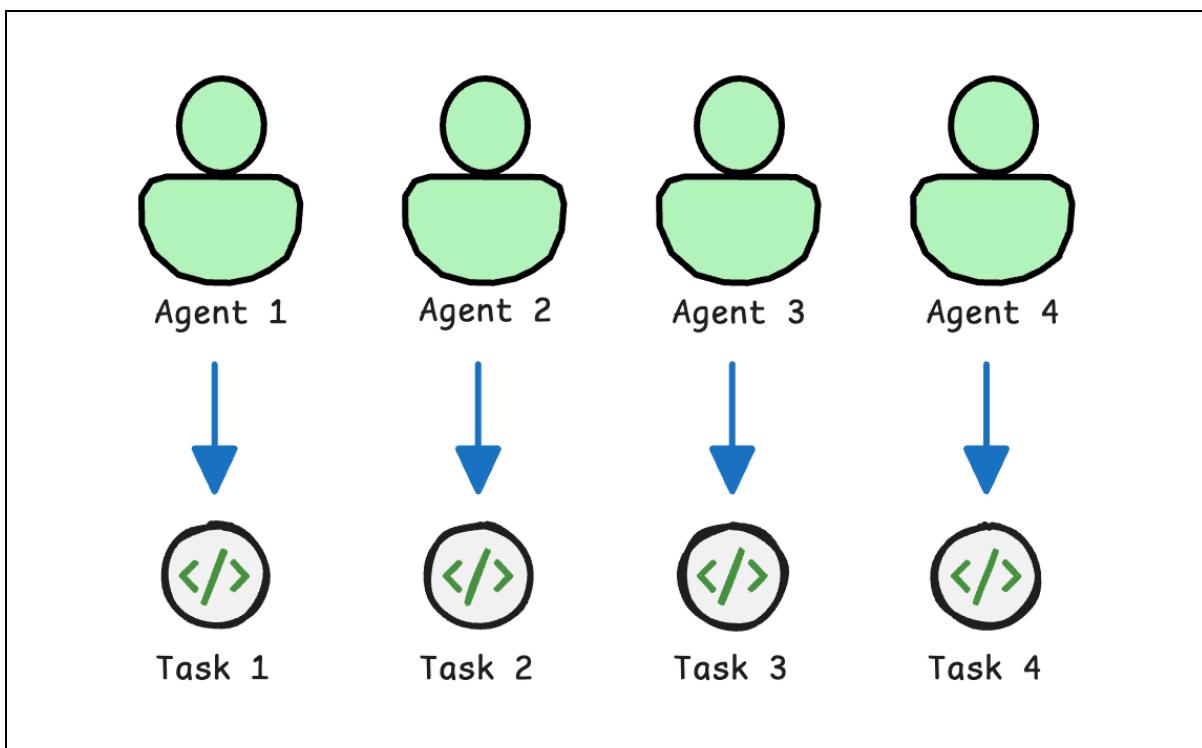
4. A Summarization Agent

- ✓ Extracts key points from articles.
- ✓ Generates concise, easy-to-read summaries.



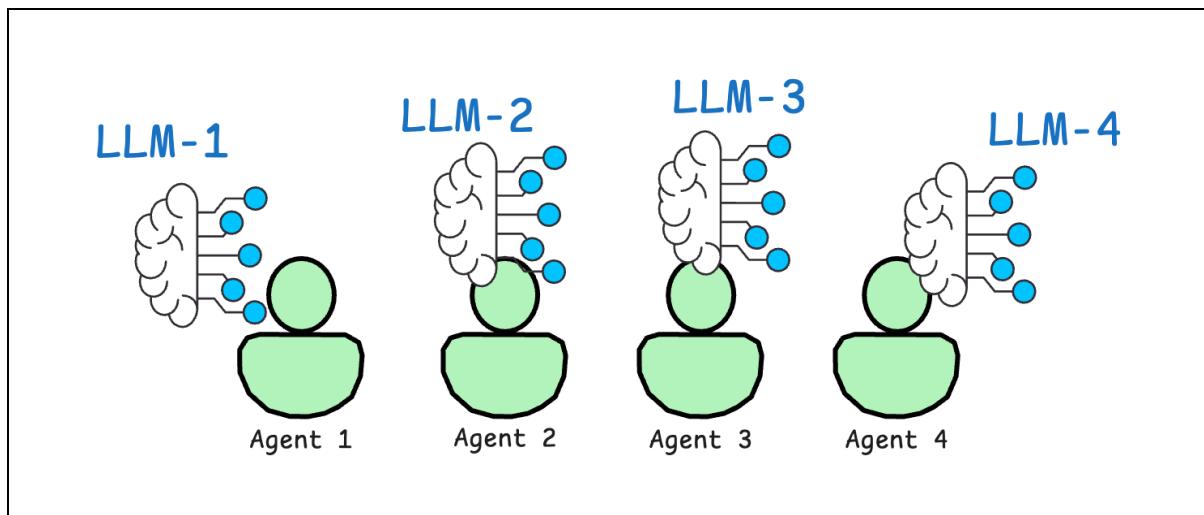
Multi-agent framework

- ✓ Multi-agent means, it's a system where multiple AI agents work together.
- ✓ Each agent has a specific role but collaborates to achieve a common goal.



Optimizing Performance with Multiple LLMs

- ✓ Each agent uses a specialized LLM for its task.
- ✓ Selecting the best LLM enhances efficiency and accuracy.



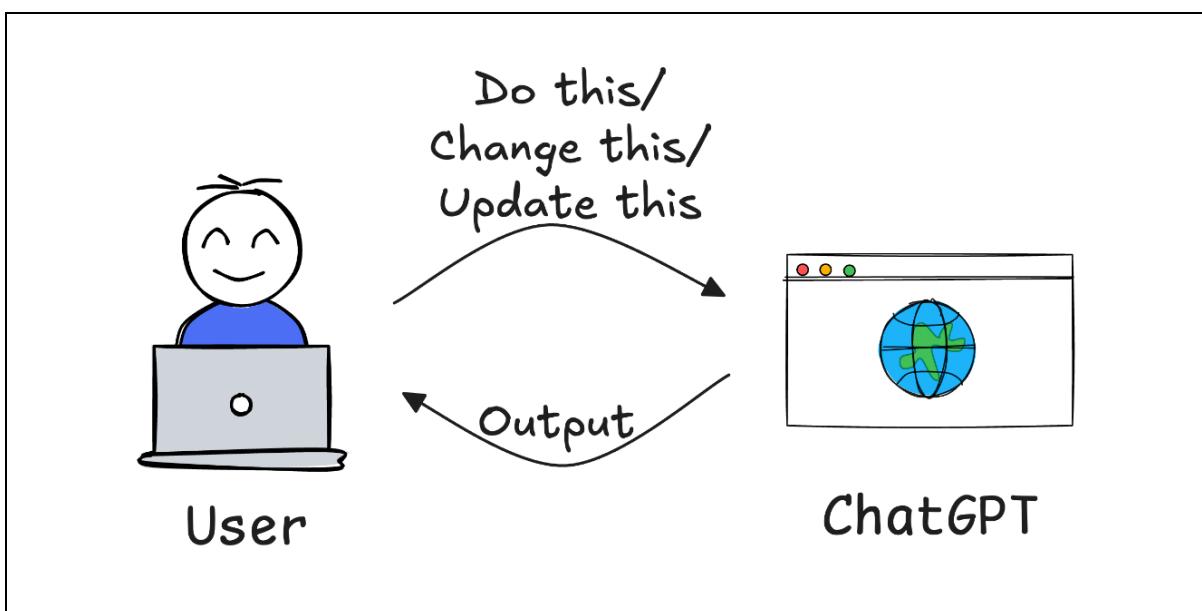
AI Agents: Learning and Adapting

- ✓ AI agents continuously learn, adapt, and optimize.
- ✓ They update automatically without manual intervention.
- ✓ Beyond fixed rules, they collaborate and solve problems autonomously.

7.2. Autonomous system perspective

Limitations of Traditional LLM Interactions

- ✓ Users must refine outputs manually.
- ✓ The process is iterative and time-consuming:
 - Ask a question.
 - Review the response.
 - Adjust the prompt.
 - Repeat until satisfied



The Challenge of Using a Standard LLM for Research

- ✓ Imagine you need a report on the latest AI research trends. With a standard LLM, you would:
 - Ask for a summary of recent AI papers.
 - Review the response and realize you need sources.
 - Request citations and get a list of papers.
 - Notice outdated sources and refine your query.
 - Repeat the process until you get a useful report.
- ✓ This iterative process takes time and effort, requiring you to make decisions at every step.

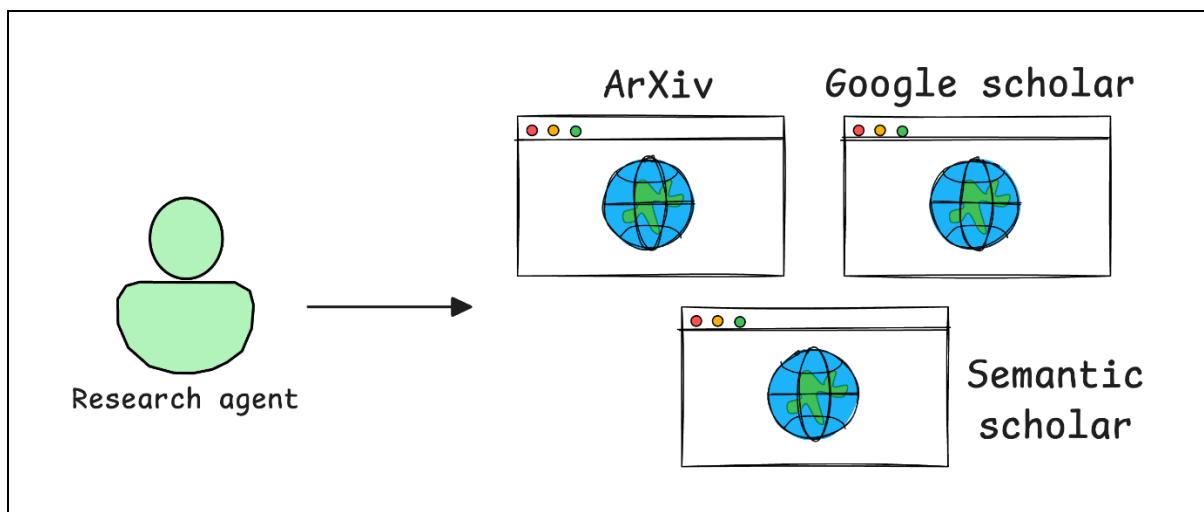
AI Agents Part - 1

AI Agents in Research

- ✓ Automate data gathering, analysis, and verification.
- ✓ Reduce the need for constant user input.
- ✓ Enhance efficiency and accuracy in research.

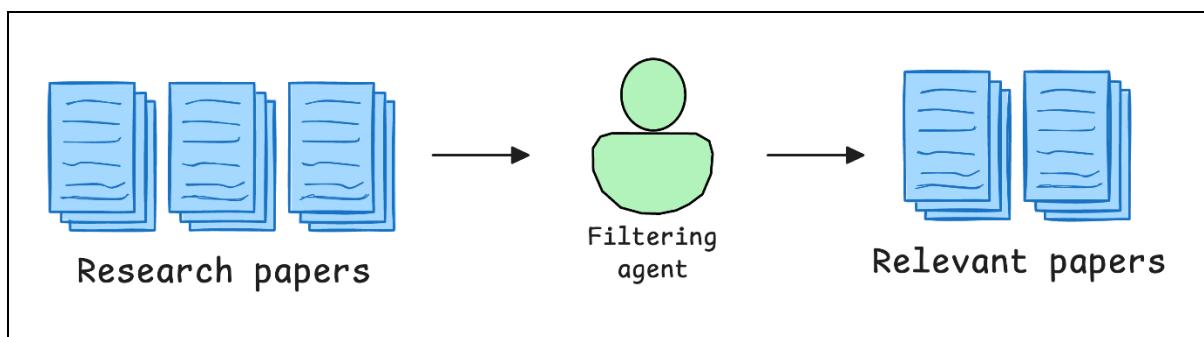
1. A Research Agent

- ✓ Searches for AI research papers automatically.
- ✓ Retrieves data from sources like arXiv, Semantic Scholar, or Google Scholar.
- ✓ Ensures access to the latest credible information.



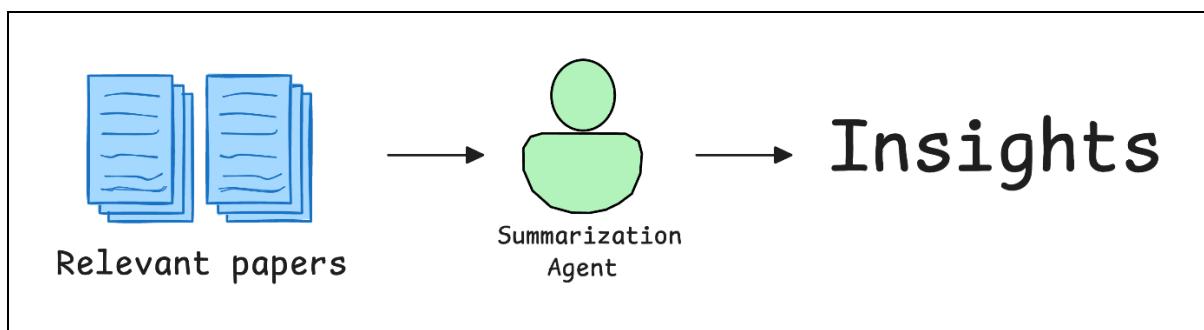
2. A Filtering Agent

- ✓ Analyzes retrieved papers for relevance.
- ✓ Filters based on citation count, publication date, and keywords.
- ✓ Ensures only high-quality research is considered.



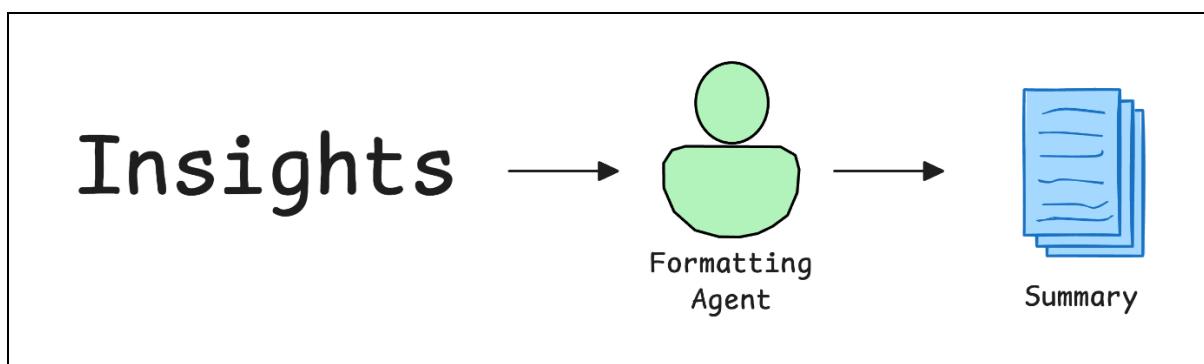
3. A Summarization Agent

- ✓ Extracts key insights from research papers.
- ✓ Compiles information into a concise, easy-to-read report.
- ✓ Saves time by presenting only the most relevant details.



4. A Formatting Agent

- ✓ Organizes the final report for clarity.
- ✓ Ensures a professional and structured layout.
- ✓ Enhances readability and presentation.



6. Core Components of AI Agents

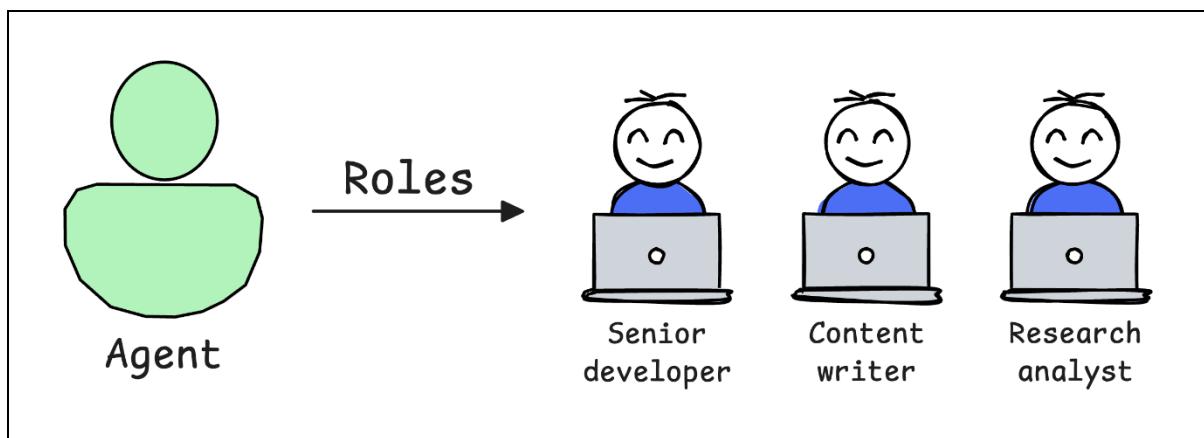
- ✓ AI agents' reason, plan, and act autonomously.
- ✓ Six key building blocks ensure reliability and intelligence.
- ✓ Essential for real-world effectiveness.

Key Elements of AI Agents

1. Role-playing
2. Focus
3. Tools
4. Cooperation
5. Guardrails
6. Memory

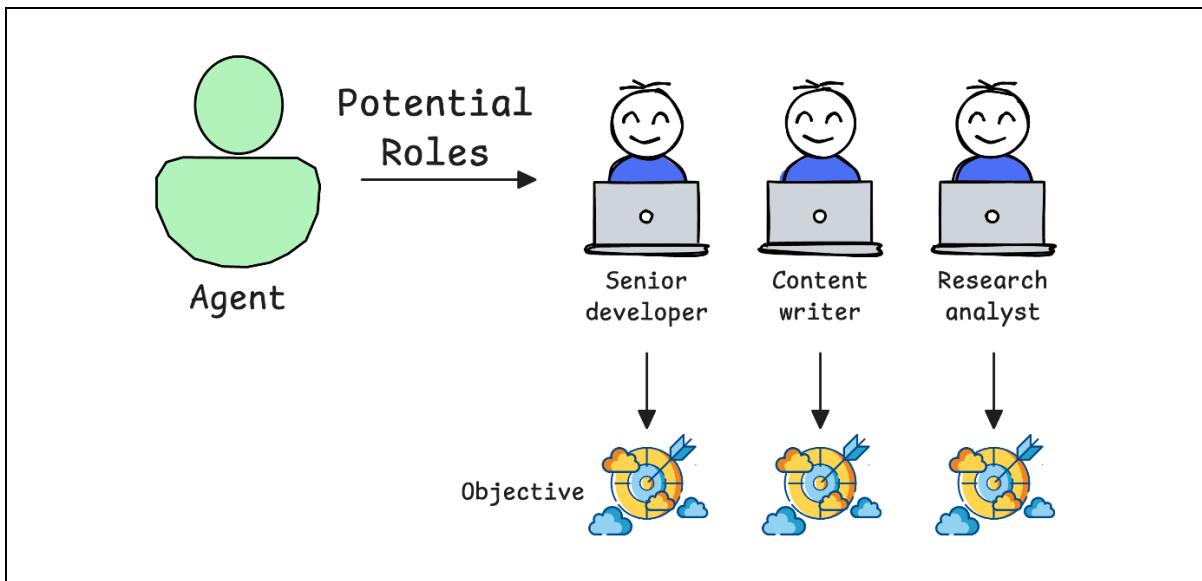
6.1. Role-Playing

- ✓ Specific roles enhance task specialization.
- ✓ Defined roles improve structured and relevant responses.
- ✓ Clear identity and objectives ensure better-aligned outputs.



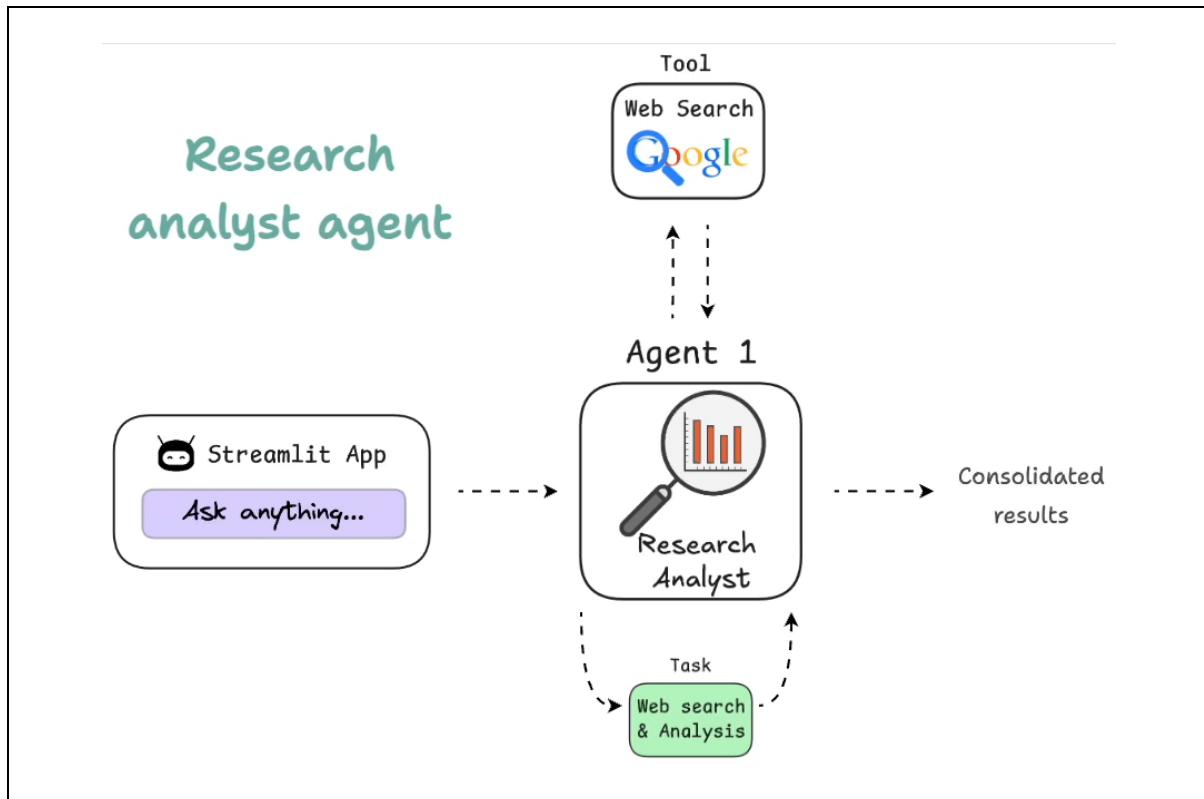
6.2. Focus

- ✓ Ensures agents stay aligned with their objectives.
- ✓ Minimizes errors and improves accuracy.
- ✓ More data isn't always better—structured input leads to better results.



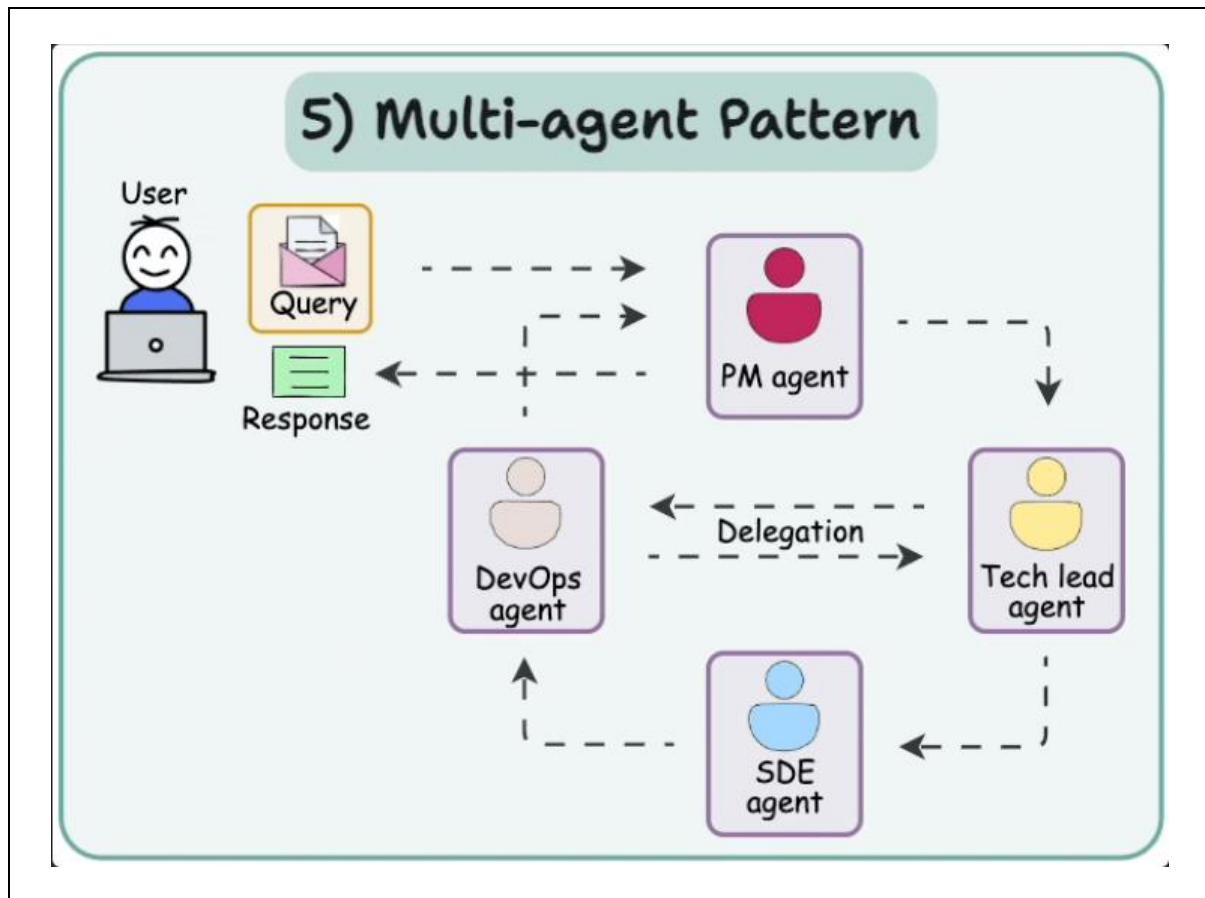
6.3. Tools

- ✓ Enhance AI agents beyond text-based reasoning.
- ✓ Enable real-time data retrieval and system interactions.



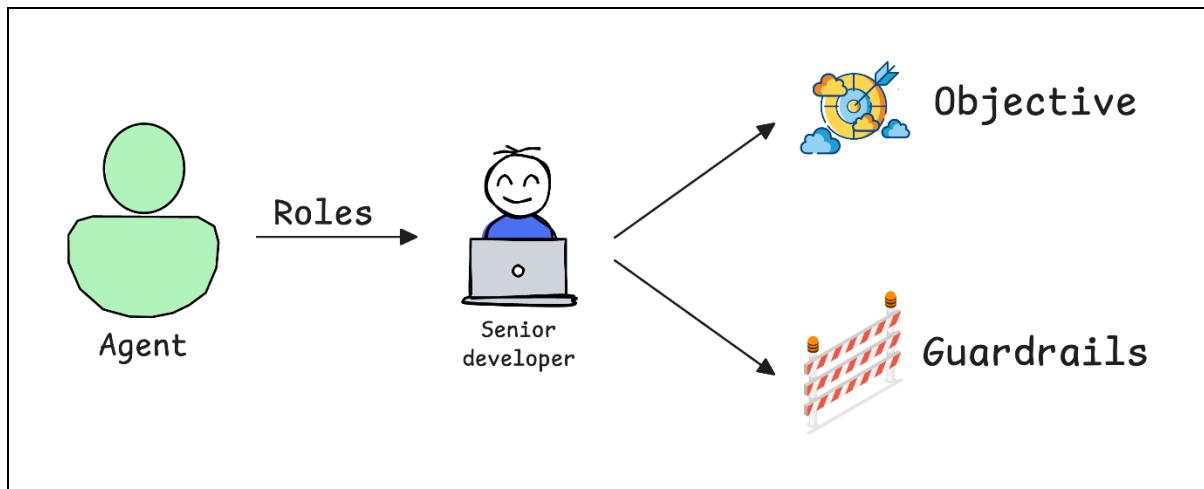
6.4. Cooperation

- ✓ Enabling Multi-Agent Collaboration
- ✓ Enables AI agents to work together efficiently.
- ✓ Enhances performance through collaboration and feedback exchange.



6.5. Guardrails

- ✓ Keeping AI Agents Reliable
- ✓ Implement safety measures for controlled behavior.
- ✓ Prevent false information, infinite loops, and unreliable decisions.

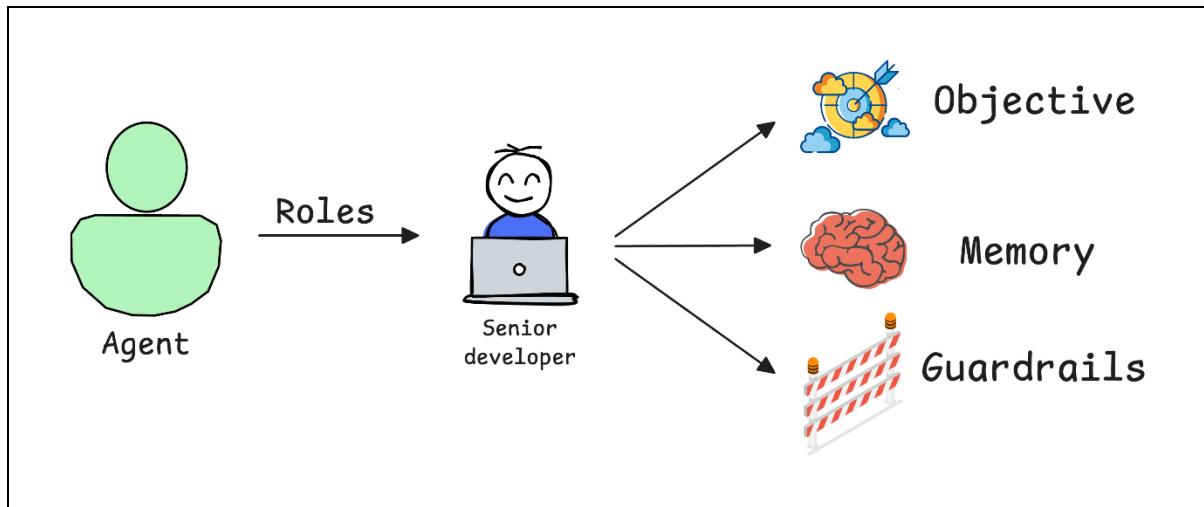


Examples of Effective Guardrails

- ✓ **Limit Tool Usage:** Prevents excessive API calls or irrelevant queries.
- ✓ **Validation Checkpoints:** Ensures outputs meet quality standards before proceeding.
- ✓ **Fallback Mechanisms:** Enables human or agent intervention when needed.

6.6. Memory

- ✓ **Without memory:** Agents reset every session, leading to repetitive interactions.
- ✓ **With memory:** Agents retain context, refine responses, and improve decision-making over time.



7. CrewAI Introduction

- ✓ It's a python framework for building multi-agent AI systems.
- ✓ By using this we can create specialized agents with unique roles, collaborate, specialize, and automate tasks.

How CrewAI Enhances AI Agents

- ✓ **Multi-Agent Collaboration:** Agents work together efficiently.
- ✓ **Specialization:** Assigns specific roles for better performance.
- ✓ **Task Automation:** Reduces manual intervention.
- ✓ **Memory & Context:** Enables continuous learning.
- ✓ **Tool Integration:** Uses external tools for enhanced capabilities.
- ✓ **Accuracy & Efficiency:** Delivers structured and reliable outputs.

Adaptive, goal-driven AI agents

- ✓ With CrewAI, we can efficiently design **adaptive, goal-driven AI agents** that collaborate and execute complex tasks independently.

8. ollama Introduction

- ✓ Ollama enables running and customizing LLMs locally.
- ✓ Avoids cloud dependency by supporting open-source AI models.
- ✓ Optimized for efficient local deployment.

How Ollama Enhances AI Agents?

- ✓ **Local Execution:** No cloud dependency, ensuring privacy.
- ✓ **Optimized Performance:** Faster inference with minimal resources.
- ✓ **Seamless Model Management:** Load, switch, and fine-tune easily.
- ✓ **Lightweight & Scalable:** Works on both edge devices and enterprises.
- ✓ **AI Agent Compatibility:** Supports autonomous agents with specialized models.

9. Creating AI Agents

- ✓ Now, let's move to implementation! Using CrewAI, an open-source framework, we can:
 - **Assign Roles:** Specialize agents for specific tasks.
 - **Set Goals:** Define clear objectives for each agent.
 - **Integrate Tools:** Equip agents with the right resources.
 - **Use Any LLM:** Seamlessly work with various language models.

10. Environment Setup

- ✓ To get started, install CrewAI as follows:

- pip install crewai
 - pip install crewai-tools

Choosing the Right LLM for CrewAI

- ✓ Ollama serves LLMs locally for seamless performance.
- ✓ CrewAI supports multiple providers, including:
 - OpenAI
 - Gemini
 - Groq
 - Azure
 - Fireworks AI
 - Cerebras
 - SambaNova & more!
- ✓ This flexibility lets you pick the best model for your needs.

Setting Up OpenAI with CrewAI

- ✓ To configure OpenAI as your LLM provider, create a `.env` file in your project's directory and add your API key like this:

- `OPENAI_API_KEY = your-api-key-here`

- ✓ Make sure to replace `your-api-key-here` with your actual OpenAI API key.
- ✓ This allows CrewAI to authenticate and interact with OpenAI's models seamlessly.

Installing and Running Ollama

- ✓ Download Ollama: Get it from official website <https://ollama.com>
- ✓ Install – Follow the on-screen setup instructions.
- ✓ Verify Installation run: `ollama --version`
- ✓ Download a Model: `ollama pull llama3.2`
- ✓ Run Locally: `ollama run llama3.2`
- ✓ Ready to Use – Serve LLMs locally without external APIs. Now, let's build our AI agents!

11. Creating Agents in CrewAI

- ✓ In CrewAI, Agents are autonomous entities with:
 - **Role:** Defines function (e.g., "Senior Technical Writer").
 - **Goal:** Specifies objective (e.g., "Write a publication-ready article").
 - **Backstory:** Adds expertise for better interactions.
- ✓ These attributes enable agents to perform tasks efficiently and independently.

12. Single-Agent vs. Multi-Agent Workflow in CrewAI

12.1. Single-Agent Workflow

- ✓ Only one agent is assigned a task.
- ✓ Handles the entire process independently.
- ✓ Simple, efficient, and best for focused tasks (e.g., writing an article).

12.2. Multi-Agent Workflow

- ✓ Multiple agents collaborate on different tasks.
- ✓ Workflows are divided (e.g., research, writing, review).
- ✓ Ideal for complex projects requiring diverse expertise.

13. AI Agents technical flow

- ✓ **Step 1:** Import Libraries.
- ✓ **Step 2:** Configure the LLM
- ✓ **Step 3:** Create Agents:
 - **Agent 1**
 - **Agent 2, etc**
- ✓ **Step 4:** Define Tasks
- ✓ **Step 5:** Create a Crew
- ✓ **Step 6:** Kickoff the Crew
- ✓ **Step 7:** Check the Response

14. Hello World Example: AI Research & Writing with CrewAI

- ✓ Simple AI workflow using CrewAI to generate a friendly greeting.
- ✓ We are using **a single-agent** workflow.

14.1. AI Agents technical flow for Simple Greeting AI using CrewAI

- ✓ **Step 1:** Import Libraries, Essential for using CrewAI and LLMs.
- ✓ **Step 2:** Configure the LLM, Define the Ollama LLaMA 3.2 language model.
- ✓ **Step 3:** Create Agent, Define its role, goal, and backstory.
 - **Agent 1:** Friendly AI, Responds with a simple greeting.
- ✓ **Step 4:** Define Task, Assign a specific task to the agent.
 - **Task:** Say "Hello, World!"
- ✓ **Step 5:** Create a Crew, Group the agent and assign the task.
- ✓ **Step 6:** Kickoff the Crew, Execute the workflow.
- ✓ **Step 7:** Check the Response, Validate and display the greeting.

Program Name Steps from 1 to 7, Simple Greeting AI using CrewAI
greetings_demo.py

```
print("Step 1: Importing the libraries")

from crewai import Agent
from crewai import Task
from crewai import Crew
from crewai import LLM

print("Step 2: Configure the llm")

llm = LLM(
    model = "ollama/llama3.2",
    base_url = "http://localhost:11434"
)

print("Step 3: Create Agent")

simple_agent = Agent(
    role = "Friendly AI",
    goal = "Respond with a simple greeting.",
    backstory = "You are a helpful AI that loves saying hello.",
    verbose = True
)

print("Step 4: Define Tasks")

greeting_task = Task(
    description = "Say 'Hello, World!'",  

    agent = simple_agent,  

    expected_output = "A friendly greeting."
)
```

```
print("Step 5: Create a Crew")

crew = Crew(
    agents = [simple_agent],
    tasks = [greeting_task],
    verbose = True
)

print("Step 6: Kickoff the Crew")

response = crew.kickoff()

print("Step 7: Check the Response")

print(response)
```

Output

Step 1: Importing the libraries
Step 2: Configure the llm
Step 3: Create Agent
Step 4: Define Tasks
Step 5: Create a Crew
Step 6: Kickoff the Crew
Step 7: Check the Response

Hello, World!

15. Use Case 1: Automated Content Creation using CrewAI

- ✓ Uses CrewAI to generate structured content.
- ✓ Senior Technical Writer (Ollama LLaMA 3.2) writes an engaging article on AI Agents.
- ✓ Ensures accuracy, efficiency, and scalability.
- ✓ We are using **a single-agent** workflow.

15.1. AI Agents technical flow for Automated content creation using CrewAI

- ✓ **Step 1:** Import Libraries - Essential for using CrewAI and LLMs.
- ✓ **Step 2:** Configure the LLM - Define the language model before creating agents.
- ✓ **Step 3:** Create Agents, Define their role, goal, and backstory.
 - Agent 1: Senior Technical Writer
- ✓ **Step 4:** Define Tasks, Assign specific tasks to each agent.
- ✓ **Step 5:** Create a Crew, assign tasks.
- ✓ **Step 6:** Kickoff the Crew, Execute the workflow.
- ✓ **Step 7:** Check the Response, Validate and analyze the output.

Program Name Steps from 1 to 7, Automated Content Creation with CrewAI
senior_tech_writer.py

```
print("Step 1: Importing the libraries")
```

```
from crewai import Agent
from crewai import Task
from crewai import Crew
from crewai import LLM
```

```
print("Step 2: Configure the llm")
```

```
llm = LLM(
    model = "ollama/llama3.2",
    base_url = "http://localhost:11434"
)
```

```
print("Step 3: Create Agent")
```

```
senior_technical_writer = Agent(
    role = "Senior Technical Writer",
    goal = """Craft clear, engaging, and well-structured
technical content based on research findings""",
    backstory = """You are an experienced technical writer
with expertise in simplifying complex concepts,
structuring content for readability, and ensuring accuracy
in documentation""",
    llm = llm,
    verbose = True
)
```

```
print("Step 4: Create Task")

writing_task = Task(
    description = """Write a well-structured, engaging,
    and technically accurate article
    on {topic}.""",
    agent = senior_technical_writer,
    expected_output = """A polished, detailed, and easy-to-
    read article on the given topic."""
)

print("Step 5: Create a Crew ")

crew = Crew(
    agents = [senior_technical_writer],
    tasks = [writing_task],
    verbose = True
)

print("Step 6: Run the Crew")

response = crew.kickoff(inputs = {"topic" : "AI Agents"})

print("Step 7: Check the Response")

print(response)
```

Output

- Step 1:** Importing the libraries
- Step 2:** Configure the llm
- Step 3:** Create Agent
- Step 4:** Create Task
- Step 5:** Create a Crew
- Step 6:** Run the Crew
- Step 7:** Check the Response

AI Agents: The Future of Intelligent Systems

Artificial intelligence (AI) has revolutionized numerous industries, transforming the way we live and work. At the heart of this technological advancement are AI agents, intelligent systems that can perform tasks autonomously, making decisions based on complex algorithms and machine learning models. In this article, we will delve into the world of AI agents, exploring their capabilities, types, and potential applications.

Understanding AI Agents

AI agents are software-based entities that operate independently, executing tasks without human intervention. They use machine learning techniques to analyze data, identify patterns, and make predictions or decisions. AI agents can be classified into two primary categories: rule-based and machine learning-based systems.

Rule-Based Systems

Rule-based systems rely on pre-defined rules and decision-making processes to guide their actions. These systems are typically used in applications where a high degree of accuracy is required, such as in healthcare and finance. Rule-based AI

AI Agents Part - 1

agents can be programmed to recognize specific patterns or anomalies, allowing them to make predictions or recommendations.

Machine Learning-Based Systems

Machine learning-based systems, on the other hand, use machine learning algorithms to analyze data and improve their performance over time. These systems are more flexible than rule-based systems, as they can adapt to changing circumstances and learn from experience. Machine learning-based AI agents are commonly used in applications such as natural language processing and computer vision.

Types of AI Agents

AI agents can be categorized into several types based on their capabilities and functionality:

1. **Simple Reflex Agents**: These agents respond to a specific set of input actions, using pre-defined rules to guide their behavior.
2. **Model-Based Reflex Agents**: These agents use knowledge-based systems to reason about the world, making decisions based on a combination of rules and machine learning models.
3. **Utility Function Agents**: These agents use utility functions to evaluate the consequences of their actions, optimizing their performance based on a set of goals or objectives.

Applications of AI Agents

AI Agents Part - 1

AI agents have numerous applications across various industries, including:

1. **Virtual Assistants**: AI agents are used in virtual assistants such as Siri and Alexa to provide personalized recommendations and perform tasks.
2. **Robotics**: AI agents control robots, allowing them to navigate complex environments and interact with humans.
3. **Healthcare**: AI agents are used in medical diagnosis and treatment, analysing patient data and providing insights to healthcare professionals.
4. **Finance**: AI agents are used in trading and investment, analysing market trends and making predictions.

Advantages of AI Agents

AI agents offer several advantages over traditional systems, including:

1. **Increased Efficiency**: AI agents can automate tasks, reducing the need for human intervention and increasing productivity.
2. **Improved Accuracy**: AI agents use machine learning algorithms to analyze data, reducing errors and improving accuracy.
3. **Personalization**: AI agents can provide personalized recommendations and services, enhancing the user experience.

Challenges and Limitations of AI Agents

While AI agents offer numerous benefits, they also pose several challenges and limitations, including:

AI Agents Part - 1

1. ****Explainability**:** AI agents can be difficult to understand, making it challenging to explain their decision-making processes.
2. ****Bias**:** AI agents can inherit biases from the data used to train them, leading to unfair outcomes.
3. ****Security**:** AI agents can pose security risks if not designed with proper safeguards.

Conclusion

AI agents are intelligent systems that have revolutionized numerous industries, transforming the way we live and work. With their capabilities in machine learning, natural language processing, and computer vision, AI agents offer numerous advantages over traditional systems. However, they also pose several challenges and limitations, including explainability, bias, and security concerns. As AI continues to evolve, it is essential to address these challenges and ensure that AI agents are designed with proper safeguards to ensure their safe and responsible deployment.

Final Thoughts

The future of AI agents holds much promise, with potential applications in numerous industries. However, as we continue to develop and deploy AI systems, it is crucial to prioritize transparency, accountability, and fairness. By doing so, we can unlock the full potential of AI agents, creating a brighter future for humanity.

In conclusion, AI agents are intelligent systems that have transformed numerous industries. With their capabilities in machine learning, natural language processing, and computer vision, they offer numerous advantages over traditional

AI Agents Part - 1

systems. However, it is essential to address the challenges and limitations associated with AI agents, ensuring that they are designed with proper safeguards to ensure their safe and responsible deployment.

As we move forward, it is crucial to prioritize transparency, accountability, and fairness in the development and deployment of AI agents. By doing so, we can unlock the full potential of these systems, creating a brighter future for humanity.

Enhancing Performance with Stronger LLMs

- ✓ Weaker LLMs may produce less accurate responses.
- ✓ Upgrade to DeepSeek or GPT-4 for better accuracy & efficiency.
- ✓ Model switching steps were covered earlier.

16. Use Case 2: Automated Healthcare AI Summary with CrewAI

- ✓ Automates research and writing on AI in healthcare.
- ✓ **Multiple agents** gather insights and creates a structured summary.
- ✓ Ensures efficiency, accuracy, and time-saving.

16.1. AI Agents technical flow for healthcare content creation using CrewAI

- ✓ **Step 1:** Import Libraries, Essential for using CrewAI and LLMs.
- ✓ **Step 2:** Configure the LLM, Define the language model before creating agents.
- ✓ **Step 3:** Create Agents, Define their role, goal, and backstory.
 - **Agent 1:** Researcher, Gathers information.
 - **Agent 2:** Writer, Summarizes research findings.
- ✓ **Step 4:** Define Tasks, Assign specific tasks to each agent.
- ✓ **Step 5:** Create a Crew, group agents and assign tasks.
- ✓ **Step 6:** Kickoff the Crew, Execute the workflow.
- ✓ **Step 7:** Check the Response – Validate and analyze the output.

AI Agents Part - 1

Program Step from 1 to 7: Healthcare use case
Name health_care_demo.py

```
print("Step 1: Importing the libraries")

from crewai import Agent
from crewai import Task
from crewai import Crew
from crewai import LLM

print("Step 2: Configure the llm")

llm = LLM(
    model = "ollama/llama3.2",
    base_url = "http://localhost:11434"
)

print("Step 3.1: Create Researcher Agent")

researcher = Agent(
    role = "Researcher",
    goal = "Gather information about a given topic.",
    backstory = "A detail-oriented AI skilled in finding
    accurate information.",
    verbose = True,
    allow_delegation = False
)
```

AI Agents Part - 1

```
print("Step 3.2: Create Writer Agent")

writer = Agent(
    role = "Writer",
    goal = "Summarize information into a concise and
    readable format.",
    backstory = "A creative AI that enjoys writing clear and
    engaging summaries.",
    verbose = True,
    allow_delegation = False
)

print("Step 4.1: Create Research Task")

research_task = Task(
    description = "Find key details about the benefits of AI in
    healthcare.",
    agent = researcher,
    expected_output = "A list of key benefits of AI in
    healthcare."
)

print("Step 4.2: Create Writing Task")

writing_task = Task(
    description = "Summarize the research findings into a
    short article.",
    agent = writer,
    expected_output = "A well-structured summary of AI's
    benefits in healthcare."
)
```

```
print("Step 5: Create a Crew")

crew = Crew(
    agents = [researcher, writer],
    tasks = [research_task, writing_task],
    verbose = True
)

print("Step 6: Run the Crew")

response = crew.kickoff()

print("Step 7: Check the Response")

print(response)
```

Output

Step 1: Importing the libraries
Step 2: Configure the llm
Step 3.1: Create Researcher Agent
Step 3.2: Create Writer Agent
Step 4.1: Create Research Task
Step 4.2: Create Writing Task
Step 5: Create a Crew
Step 6: Run the Crew
Step 7: Check the Response

The Transformative Benefits of AI in Healthcare

Artificial Intelligence (AI) is revolutionizing the healthcare sector through a range of innovative applications that enhance diagnostic accuracy, personalize medicine, and streamline operations. Here are some key benefits of AI in healthcare that demonstrate its transformative potential:

1. **Enhanced Diagnostic Accuracy**: AI algorithms, especially those utilizing deep learning techniques, surpass traditional diagnostic methods by providing more accurate analyses of medical images, such as X-rays, MRIs, and CT scans. Research indicates that AI can often match or even outperform human radiologists in identifying diseases, including cancer.
2. **Personalized Medicine**: By analysing extensive patient data, AI helps in crafting personalized treatment plans. It predicts individual responses to various therapies based on genetic profiles, lifestyle factors, and disease characteristics, resulting in more tailored and effective healthcare.
3. **Predictive Analytics**: AI's ability to process large datasets enables it to recognize patterns and forecast patient outcomes. Machine learning models can predict disease outbreaks or assess which patients are at a higher risk of readmission, facilitating timely preventive care.
4. **Operational Efficiency**: AI enhances healthcare operations through automation of routine tasks such as patient scheduling, flow management, and billing processes. This efficiency not only streamlines healthcare delivery but also alleviates administrative workloads on healthcare staff.
5. **Drug Discovery and Development**: The integration of AI in drug discovery accelerates the identification of potential drug candidates and assesses their efficacy by analysing biological data. This significantly reduces the time and financial investment needed to bring new medications to market.
6. **Remote Monitoring and Telemedicine**: AI-powered tools allow for continuous health monitoring through wearable devices. These applications enable healthcare providers to track

AI Agents Part - 1

vital signs and address emerging health issues early on, particularly beneficial for managing chronic illnesses.

7. ****Enhanced Patient Engagement**:** Virtual assistants and chatbots driven by AI offer patients immediate access to information and support, fostering better communication and engagement with healthcare providers. This assists patients in managing conditions effectively and adhering to treatment plans.
8. ****Integration of Data from Multiple Sources**:** AI systems excel at synthesizing health data from diverse sources like electronic health records (EHRs), genetic data, and wearables. This comprehensive analysis equips healthcare professionals with a holistic view for informed decision-making.
9. ****Efficiency in Clinical Trials**:** AI optimizes various aspects of clinical trials, including design, patient recruitment, and monitoring, thereby expediting the overall process. It helps in identifying suitable candidates and predicting responses, which enhances the likelihood of trial success.
10. ****Supporting Healthcare Professionals**:** By helping with administrative tasks and processing complex medical data, AI supports healthcare providers in focusing more on their patients. It offers evidence-based recommendations that improve the quality of care.

In summary, the applications of AI in healthcare showcase its incredible ability to improve diagnostic processes, enhance patient outcomes, and create operational efficiencies. As AI continues to evolve, it holds the promise of transforming healthcare delivery into a more accurate, efficient, and patient-centered system.

17. Use Case 3: Multi-Agent Research Assistant with CrewAI

- ✓ Uses CrewAI to automate and enhance research tasks.
- ✓ **Multiple Agents**
 - Research Agent gathers real-time, relevant information.
 - Summarization Agent structures findings into a clear summary.
 - Fact-Checking Agent verifies accuracy for reliability.
 - Improves accuracy, efficiency, and scalability compared to a single-agent system

17.1. AI Agents technical flow for Multi-Agent Research Assistant with CrewAI

- ✓ **Step 1:** Import Libraries – Load CrewAI, dotenv, and SerperDevTool.
- ✓ **Step 2:** Load Environment & Enable Web Search – Use `load_dotenv()` and SerperDevTool for real-time search.
- ✓ **Step 3:** Create Agents – Define roles and goals.
 - **Step 3.1:** Research Agent – Gathers the latest information using Serper Dev Tool.
 - **Step 3.2:** Summarization Agent – Extracts and summarizes key insights.
 - **Step 3.3:** Fact-Checking Agent – Verifies and corrects inaccuracies.
- ✓ **Step 4:** Define Tasks – Assign tasks to agents.
 - **Step 4.1:** Research Task – Use SerperDevTool to gather insights.
 - **Step 4.2:** Summarization Task – Convert findings into a structured summary.
 - **Step 4.3:** Fact-Checking Task – Validate and correct the summary.
- ✓ **Step 5:** Create a Crew – Group agents, assign tasks, and set sequential execution.
- ✓ **Step 6:** Run the Crew – Execute the workflow with an input topic.
- ✓ **Step 7:** Check the Response – Display the fact-checked, verified summary.

Enabling Web Search

- ✓ Integrate Serper.dev for real-time web searches.
 - Get a free API key from Serper.dev.
 - Set the API key in the .env file for authentication.
- ✓ Next step: Define agents and their tasks!

```
OPENAI_API_KEY="sk-38381...."
```

```
SERPER_API_KEY="96f29..."
```

AI Agents Part - 1

Program Name Steps from 1 to 7, Multi-Agent Research Assistant with CrewAI
multi_agent_demo.py

```
print("Step 1: Importing the libraries")

from crewai import Agent, Task
from dotenv import load_dotenv
from crewai import Crew, Process

from crewai_tools import SerperDevTool

print("Step Special 1: Load Environment & Enable Web Search –
      Use load_dotenv() and SerperDevTool for real-time search.")

load_dotenv()

serper_dev_tool = SerperDevTool()

print("Step 2: Configure the llm: Using default LLM now")

print("Step 3.1: Research Agent ")

research_agent = Agent(
    role = "Internet Researcher",
    goal = "Find the most relevant and recent information
          about a given topic.",
    backstory = """You are a skilled researcher, adept at
                  navigating the internet and gathering high-quality,
                  reliable information.""",
    tools = [serper_dev_tool],
    verbose = True
)
```

AI Agents Part - 1

```
print("Step 3.2: Summarization Agent")

summarizer_agent = Agent(
    role = "Content Summarizer",
    goal = "Condense the key insights from research into a
short and informative summary.",
    backstory = """You are an expert in distilling complex
information into concise, easy-to-read summaries.""",
    verbose = True
)

print("Step 3.3: Fact-Checking Agent")

fact_checker_agent = Agent(
    role = "Fact-Checking Specialist",
    goal = "Verify the accuracy of information and remove
any misleading or false claims.",
    backstory = """You are an investigative journalist with a
knack for validating facts,
ensuring that only accurate information is published."""",
    tools = [serper_dev_tool],
    verbose = True
)
```

```
print("Step 4.1: Research Task")

research_task = Task(
    description = """Use the SerperDevTool to search for the
    most relevant and recent data about {topic}."""
    "Extract the key insights from multiple sources.",
    agent = research_agent,
    tools = [serper_dev_tool],
    expected_output = "A detailed research report with key
    insights and source references."
)

print("Step 4.2: Summarization Task")

summarization_task = Task(
    description = "Summarize the research report into a
    concise and informative paragraph. "
    "Ensure clarity, coherence, and completeness.",
    agent = summarizer_agent,
    expected_output = "A well-structured summary with the
    most important insights."
)

print("Step 4.3: Fact-Checking Task")

fact_checking_task = Task(
    description = "Verify the summarized information for
    accuracy using the SerperDevTool. "
    "Cross-check facts with reliable sources and correct any
    errors.",
    agent = fact_checker_agent,
    tools = [serper_dev_tool],
    expected_output = "A fact-checked, verified summary of
    the research topic."
)
```

```
print("Step 5: Create a Crew")

research_crew = Crew(
    agents = [research_agent, summarizer_agent,
              fact_checker_agent],
    tasks = [research_task, summarization_task,
              fact_checking_task],
    process = Process.sequential,
    verbose = True
)

print("Step 6: Run the Crew")

result = research_crew.kickoff(inputs={"topic": "The impact of AI
on job markets"})

print("Step 7: Check the Response")

print("\nFinal Verified Summary:\n", result)
```

Output

- Step 1:** Importing the libraries
- Step Special 1:** Load Environment & Enable Web Search – Use load_dotenv() and SerperDevTool for real-time search
- Step 2:** Configure the llm: Using default LLM now
- Step 3.1:** Research Agent
- Step 3.2:** Summarization Agent
- Step 3.3:** Fact-Checking Agent
- Step 4.1:** Research Task
- Step 4.2:** Summarization Task
- Step 4.3:** Fact-Checking Task
- Step 5:** Create a Crew
- Step 6:** Execute the Crew

Step 7: Check the Response

Final Verified Summary:

The impact of Artificial Intelligence (AI) on job markets is significant, reshaping job roles and enhancing productivity. With AI automating traditional positions, particularly in technical fields, new job categories are emerging, forecasted to create 20 to 50 million jobs globally. AI is expected to boost workplace efficiency by up to 40%. However, this transition will vary across industries, with notable increases in AI job listings and demand for AI-literate professionals. Policymakers are urged to implement proactive measures, as AI could affect 40% of jobs worldwide, necessitating a fair transition into an AI-driven economy. Education will play a crucial role in preparing the workforce, emphasizing specialized skills in technology and AI. Overall, while AI presents challenges, it also offers opportunities for economic growth and workforce development if adequately managed.