AI Agent 과정

crewAI

목차 crewAI

1. crewAI 소개 및 사용법

왜 Multi-Agent 시스템이 필요한가

Single Agent 한계:

- 여러개의 Tool을 가졌더라도 단일 Agent로는 해결 불가능한 작업 존재
- 인지 과부하 (Cognitive Overload):
  - 하나의 LLM이 리서치, 데이터 분석, 초안 작성, 교정/편집 등 모든 역할을 동시에 수행
  - 프롬프트가 지나치게 복잡해지고 각 단계의 맥락을 놓침

왜 Multi-Agent 시스템이 필요한가

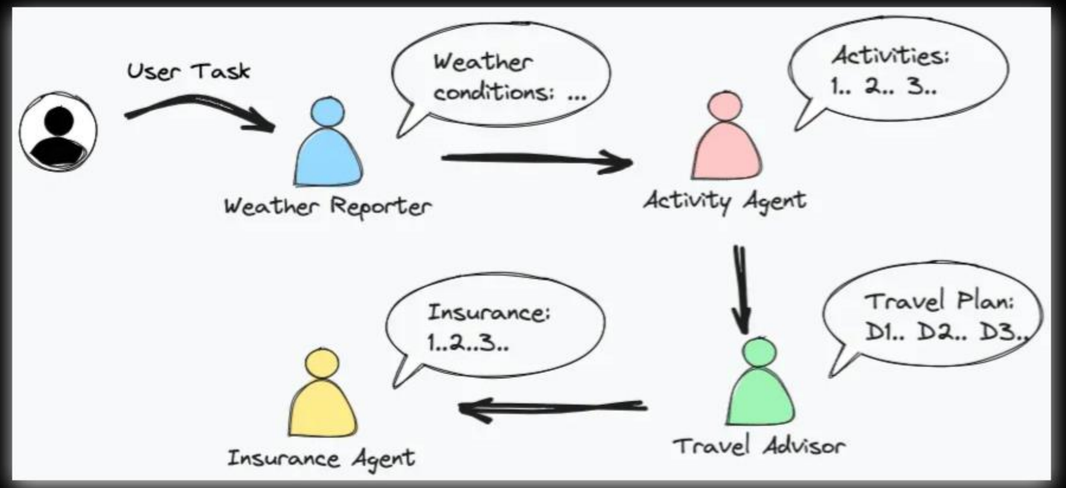
Single Agent 한계:

- 전문성 부족:
  - 리서처 능력과 작가 능력은 서로 다른 사고방식과 스킬셋
  - 하나의 Agent에게 두 역할을 모두 요구
  - 각 역할의 결과물 품질이 저하
- 프롬프트 오염 (Prompt Contamination):
  - LLM이 여러 역할을 동시에 수행하려다 보면, 각 역할의 사고방식이 서로 섞임
  - 리서처의 객관적인 톤과 전략가의 창의적인 톤이 충돌하여 이도저도 아닌 결과물 생성

CrewAI

Master < Crew

- 하나의 AI에게 모든 것을 맡기는 대신, 각자의 역할에 특화된 여러 AI Agent들이 협력
- 특화 에이전트:
  - '리서처', '데이터 분석가', '보고서 작성가' 등 특정 도메인에 고도로 특화된 AI Agent를 각각 활용
- 감독(Orchestration) AI:
  - 크루의 작업을 조율하고 지휘하는 프로젝트 매니저
  - CrewAI에서는 'Process'
- 단계별 검증:
  - 각 에이전트가 역할을 마칠 때마다 결과물을 검토
  - 최종 결과물의 완성도를 점진적으로 높임



CrewAI

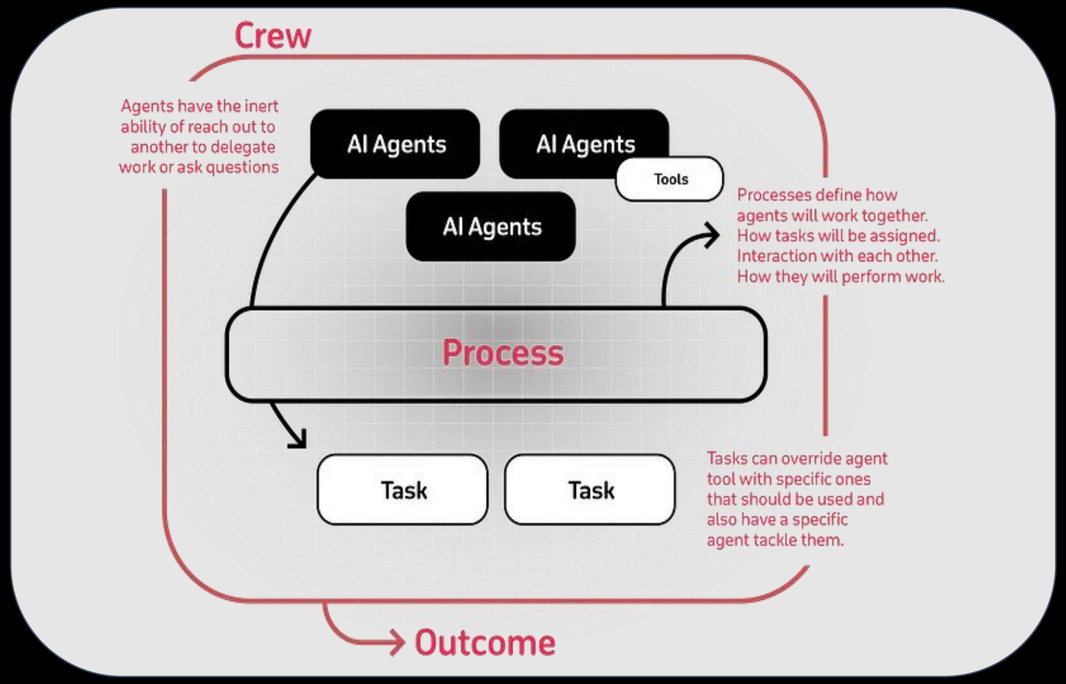
CrewAI:

- 정교한 Multi-Agent 시스템을 구축하기 위해 설계된 오픈소스 프레임워크
  - 개발자가 복잡한 오케스트레이션 로직을 직접 구현할 필요 없음
  - 각 Agent의 역할과 작업을 정의하는 것만으로 협업 프로세스를 자동화
- 역할 기반 설계 (Role-Playing):
  - 각 Agent에게 명확한 역할(Role), 목표(Goal), 배경(Backstory)을 부여
- 자율적 위임 (Autonomous Delegation):
  - Agent들이 필요에 따라 서로에게 작업을 위임하거나 질문
- 유연한 프로세스 관리:
  - 작업 흐름을 순차적(Sequential) 또는 계층적(Hierarchical)으로 손쉽게 정의

CrewAI

CrewAI 4대 핵심 요소:

- Agents: 업무를 수행하는 팀원
- Tasks: Agent에게 할당되는 구체적인 업무 지시서
- Crew: Agents와 Tasks를 모아놓은 팀
- Process: 팀원들이 어떤 방식으로 협업할지를 정의하는 업무 규칙



CrewAI의 4대 핵심 요소

Agent: 업무 수행 주체

- Role
- Goal
- Backstory

```
researcher = Agent(
    role='선임 기술 리서처',
    goal='특정 기술 주제에 대한 최신 정보를 웹에서 찾아 요약합니다.',
    backstory=(
        '당신은 기술 전문 매체에서 15년간 근무한 베테랑 기자입니다. '
        '항상 사실에 기반한 정보만을 신뢰하며, 여러 출처를 교차 검증하여 '
        '가장 정확한 내용을 찾아내는 데 특화되어 있습니다.'
    ),
```

CrewAI의 4대 핵심 요소

Agent: 업무 수행 주체

- Role
  - 해당 Agent가 담당할 구체적인 업무 영역과 전문 분야를 명확히 정의
  - 다른 Agent와 구별되는 고유한 전문성과 책임 범위를 설정
- Goal
  - Agent의 모든 행동과 판단의 기준이 되는 구체적이고 측정 가능한 목표
  - 성공 여부를 객관적으로 평가할 수 있는 명확한 성과 지표 포함
- Backstory
  - Agent의 의사결정 스타일과 문제 해결 접근 방식을 결정하는 배경 정보
  - 해당 분야에서의 경험과 축적된 지식이 업무 수행에 어떻게 반영될지 설명

CrewAI의 4대 핵심 요소

Task:

- Agent가 수행해야 할 업무를 상세하게 기술한 지침
- Description
- Expected Output
- Agent Assignment:

```
research_task = Task(
  description=(
      'AI Agent 프레임워크인 "CrewAI"와 "LangGraph"의 '
      '핵심적인 차이점 3가지를 찾아서 비교 설명해주세요.'),
  expected_output=(
      '각 프레임워크의 장단점을 포함한 비교 분석 보고서. '
      '보고서는 마크다운 형식으로 작성되어야 하며, '
      '최소 500자 이상이어야 합니다.'),
  agent=researcher # 이 Task는 'researcher' Agent가 담당합니다.)
```

CrewAI의 4대 핵심 요소

Task:

- Agent가 수행해야 할 업무를 상세하게 기술한 지침
- Description:
  - 수행할 작업의 범위, 방법, 주의사항 등을 애매함 없이, 실행 가능한 수준으로 명확하게 기술
- Expected Output:
  - 작업 완료 시 기대하는 결과물의 형태, 분량, 품질 수준을 명확히 정의
  - 후속 Task가 이 결과물을 입력으로 사용할 수 있도록 하는 '데이터 규격' 역할
- Agent Assignment:
  - 해당 업무에 가장 적합한 전문성을 보유한 Agent를 선정
  - Agent의 역량과 현재 업무 부하를 고려한 효율적 업무 배분
  - 필요시 여러 Agent의 협업이나 단계별 담당자 변경 계획 수립

CrewAI의 4대 핵심 요소

Crew

- Agent들의 집합
  - 공통 목표 달성을 위해 상호 보완적인 전문성을 가진 Agent들의 조합
  - 팀 내 각 구성원의 역할과 책임이 중복되지 않고 시너지를 창출하는 구조
  - 프로젝트 특성에 맞는 최적의 팀 구성으로 효율성과 품질 동시 추구
- 목표 달성
  - 각 Agent의 개별 성과가 전체 프로젝트 목표에 기여하는 구조적 연결
  - 부분 최적화가 아닌 전체 최적화 관점에서의 팀 성과 추구
  - 최종 목표 달성도를 객관적으로 측정하고 평가할 수 있는 시스템 구축

CrewAI의 4대 핵심 요소

Crew

- Process 관리
  - 작업 간 의존성과 우선순위를 고려한 효율적 업무 진행 순서 설정
  - 각 단계별 완료 기준과 다음 단계로의 전환 조건 명확히 정의
  - 예상치 못한 상황이나 지연 발생 시 대응할 수 있는 유연한 프로세스 구축

CrewAI의 4대 핵심 요소

Process

- Sequential
  - 이전 단계의 결과물이 다음 단계의 입력 자료가 되는 순차적 진행 방식
  - 각 단계별 품질 검증과 승인 절차를 통한 단계적 품질 보장 시스템
  - 명확한 업무 흐름으로 진행 상황 파악과 책임 소재 명확화 가능
- Hierarchical
  - 매니저 Agent가 전체 프로젝트를 조망하며 최적의 업무 분배와 일정 관리
  - 복잡한 프로젝트에서 발생할 수 있는 우선순위 충돌이나 자원 배분 문제 해결
  - 의사결정 지원 및 피드백으로 프로젝트 진행 속도와 품질 향상

```
my_crew = Crew(
  agents=[researcher, writer], # 팀원으로 두 Agent를 지정
  tasks=[research_task, write_task], # 수행할 Task 목록
  process=Process.sequential # 순차적으로 업무를 진행하도록 설정)
```

Caveats

CrewAI의 한계

- 제한된 유연성:
  - Sequential, Hierarchical 등 정해진 프로세스 외, 복잡한 조건부 분기나 동적인 루프 구현 어려움
- 디버깅의 어려움:
  - 추상화 수준이 높은 만큼, Agent가 왜 특정 행동을 했는지 그 내부의 상세한 생각 흐름을 추적하고 디버깅하기가 LangGraph에 비해 상대적으로 어려움
- 비용:
  - 여러 Agent가 자율적으로 여러 번 LLM을 호출하므로, 의도치 않은 대량의 API 비용 발생