

Agents: An Open-source Framework for Autonomous Language Agents

에이전트

- 특정 목적을 달성하기 위해 설계된 프로그램
- 일정한 범위 내에서 자율적으로 동작할 수는 있지만 완전하지 않고 외부로부터 입력을 받아 정해진 규칙에 따라 동작한다.

자율 에이전트란?

- 자기 환경을 인식하고 이를 바탕으로 자율적으로 행동하거나 결정을 내리는 소프트웨어 or 하드웨어
- 자신의 목표를 설정하고 이를 달성하기 위해 스스로 학습하고, 행동하고, 결정을 내린다.

Autonomous Language Agents

- 자연어 처리를 수행하는 에이전트
- 인간의 개입 없이 스스로 학습하고 판단하여 행동하는 언어 처리 시스템 즉 자연어처리에 초점을 맞춘 자율 에이전트라고 생각하면 된다.
- 즉 독립적으로 언어처리 작업을 수행할 수 있고 언어를 이해하고 생성하는 능력을 가진다. 보통 대화형 시스템, 번역 시스템, 텍스트 생성 또는 분석 시스템 등에 사용된다.

▼ AGENTS 기능

AGENTS

- 최첨단 자율 언어 에이전트를 구축하도록 도와주는 라이브러리
- 언어 에이전트를 위한 통합 프레임워크이자 오픈 소스 라이브러리

목표

- 비전문가도 쉽게 접근할 수 있는 라이브러리를 만드는 것
- 언어 에이전트의 배포 및 공유를 더 쉽게 만드는 것이 목표

- 개발자가 언어 에이전트를 사용하여 애플리케이션을 구축하고, 연구자가 언어 에이전트 연구를 수행하며, 기술 전문가가 아닌 일반 사용자가 개인화된 언어 에이전트를 구축하고 사용자 지정할 수 있도록 지원하는 것을 목표

기존 문제점

- 숙련된 개발자나 연구자들도 새로운 에이전트를 customizing, tuning, and deploying 하는것이 쉽지 않다
- 언어 에이전트의 가능성을 보여주는 개념 증명이지 대규모 프레임워크가 아니기 때문에 언어 에이전트를 구축하고 서스터마이징 하는데 사용되기 어렵다.
- 기존 언어 에이전트 프레임워크들의 경우 짧은 ask description에 의존하고 계획과 행동이 LLM에 의존하기 때문에 실행 마다 결과가 다르게 나오는 무작위성을 초래하여 언어 에이전트를 조정하는것을 어렵게 한다.

제공 기능

▼ Long-short term memory

- 메모리 구성요소를 통합
- VectorDB와 시맨틱 검색을 사용하여 장기 메모리를 저장하고 검색
- 스크래치패드(임시 정보 저장하는 메모리 공간)를 사용하여 단기 워킹 메모리(일시적으로 정보 저장하는 메모리)를 정기적으로 업데이트

▼ Tool usage & Web navigation

- 외부 도구를 사용하고 인터넷 서핑을 할 수 있는 기능
- 외부 API를 지원하고 다른 도구를 쉽게 통합 할 수 있는 추상 클래스 제공

▼ Multi-agent communication

- 동적 스케줄링이라는 새로운 기능 지원
- 동적 스케줄링은 컨트롤러 에이전트를 정의 하고 현재 히스토리를 고려하여 다음 작업을 수행할 에이전트를 결정

▼ Human-agent interaction

- 단일 , 다중 에이전트 시나리오에서 사람과 에이전트 간의 상호작용을 원활하게 지원

- 기존에는 인간 사용자와 에이전트가 상호작용하는것이 어려웠다.

▼ Controllability

▼ 자율 언어 에이전트

자율 언어 에이전트

Auto- GPT(Richards 등, 2023), BabyAGI(Nakajima, 2023),
SuperAGI(SuperAGI, 2023)

언어 에이전트 프레임워크

- Transformers Agents [Wolf et al., 2020]
- LangChain [LangChain, 2022]

Table 1: Comparison of Language Agent Frameworks

Framework	Tool Usage	Long-short Term Memory	Multi-Agent	Human-Agent Interaction	Symbolic Control
Transformers Agents	✓	✗	✗	✗	✗
LangChain	✓	✓	✗	✗	✗
Auto-GPT	✓	✗	✗	✗	✗
Gentopia	✓	✗	✗	✗	✗
XLang	✓	✗	✗	✗	✗
Meta-GPT	✓	✗	✓	✗	✗
Camel	✓	✗	✓	✗	✗
AgentVerse	✓	✓	✓	✓	✗
AGENTS	✓	✓	✓	✓	✓

다른 Language Agent Frameworks과 비교해보았을때 도구 사용, 장단기 메모리, 다
중 에이전트 통신 기능을 동시에 지원하는 프레임 워크는 AGENTS가 유일하다는것을
확인 할 수 있다.

또한 처음으로 SOP를 통해 사람과 에이전트간의 상호 작용 및 제어를 제공한다.

▼ Library Design

Code 1: Exemplar code for initializing and running a (multi) agent system with AGENTS

```

1 def main():
2     # agents is a dict of one or multiple agents.
3     agents = Agent.from_config("./config.json")
4     sop = SOP.from_config("./config.json")
5     environment = Environment.from_config("./config.json")
6     run(agents, sop, environment)

```

에이전트, 환경, SOP 주요 3가지 클래스를 초기화

구성파일은 핵심 개체를 정의하고 복잡한 prompts를 모듈화된 prompts로 팩터화 한다.

Code 2: Exemplar code for the running loop of a (multi) agent system in AGENTS

```

1 def run(agents, sop, environment):
2     while not sop.finished:
3         agent, state = sop.step(agents, environment)
4         action = agent.step(state, environment)
5         environment.update(agent, action)
6         # optional, in case of dynamic planning
7         # new_states = get_new_states(action)
8         # sop.add_states(new_states)

```

먼저 상태를 전환 하고 에이전트와 환경에 따라 다음에 행동할 에이전트 선택, 그 후 자신의 상태와 환경에 따라 ACTION을 취한다.

그 후 새로운 행동에 따라서 환경을 업데이트 해주고 만약 중간에 실행 결과에 따라 계획을 동적으로 조정해야하는 경우 작업 출력을 파싱하여 새로운 상태를 정의하고 현재의 SOP에 추가 할 수 있다.

Agent class

- agent._observe(environment) : 장단기 메모리를 유지하며 환경을 관찰
- agent._act() : 현재 상태에 따라 행동
- agent._update_memory(): 메모리를 업데이트
- agent.step() : 모든 메서드가 래핑
- _is_human 속성: True일 경우 인간 사용자에게 관찰 및 메모리 정보를 제공하고 입력을 받는다. 이를 통해 인간 사용자가 언어 에이전트 역할을 수행하게 한다.

SOP

- SOP에 설명된 작업을 수행할때 그에 맞게 상태가 변경된다. 즉 하위 작업 또는 하위 목표를 지정해주는것이다.

- 상태는 State 클래스로 추상화되고 State 클래스에 상태에 따라서 사용할 수 있는 다양한 도구 와 API가 포함되어있다. 이때
- Component 클래스의 경우 상태에서 작업에 사용할 수 있는 모든것을 추상화한다.
- Component 클래스의 경우 "PromptComponent"와 "ToolComponent" 이렇게 2가지의 하위 클래스를 가진다.
- "PromptComponent"는 작업/목표, 규칙/제약 조건, 상황별 학습을 위한 (단계별) 데모, 출력 형식을 지정하는 모듈화된 프롬프트가 포함되어 있다.
- ToolComponent는 모듈화된 프롬프트 외에도 웹 검색, 지식 베이스 등의 외부 도구 및 API를 포함한다.
- `sop._transit()` : 상태 전환 함수
- `sop._route()`: 상담원 라우팅 함수
- 이 둘은 `sop.next()`로 래핑

Environment 클래스

- 에이전트가 위치한 환경을 추상화
- `environment._observed()`: 환경이 에이전트 행동에 어떻게 영향을 미치는지 정의
- `environment.update()`: 에이전트의 행동이 환경에 영향을 미치는지

핵심 기능의 구현 세부 사항

장단기 기억

- Zhou 등에 따라 구현
- 장기 기억은 문장을 인베딩 하고 벡터DB에 저장되며, 시맨틱 검색을 통해 쿼리되며 단기 기억의 경우 자연어 형태로 되어 있고 프롬프트를 통해 LLM에 의해 업데이트 된다.

도구 사용 및 웹 탐색

- `ToolComponent.func()`: API 호출
- API 호출이 컨텍스트에 따라 달라지는 복잡한 도구라면 OpenAI의 GPT API의 '함수 호출' 기능을 통합하여 LLM이 도구 사용 방법을 결정
- 웹 탐색은 전문 도구로 구현

다중 에이전트 통신

- 이전 동작, 환경 및 현재 상태의 대상을 고려하여 다음 동작을 수행할 에이전트를 결정하는 컨트롤러 기능을 포함한다.

인간-에이전트 상호 작용

- 에이전트 시스템에서 인간과 에이전트 간의 상호 작용을 지원

배포

- FastAPI를 통해 언어 에이전트를 API로 배포(실제 애플리케이션에 언어 에이전트를 통합 용이)
- 기존 오픈소스 프레임 워크는 터미널이나 Gradio에서 실행

AGENTS

- 언어 에이전트의 배포 및 공유를 더 쉽게 만드는것이 목표

상당원 허브

- 사용자들이 만든 언어 에이전트를 공유하고 검색하여 다운로드 할 수 있는 플랫폼
- 이를 통해 다른 사람이 만든 에이전트를 커스터 마이징 할 수 있게 했다.

에이전트 시스템 자동 생성

- 사용자가 일일이 SOP를 수동으로 지정하는것은 번거롭다.
- 검색 증강 생성(RAG)을 기반으로 SOP 자동 생성 기능을 제공한다.

사례

단일 상당원 시스템

채팅 봇, 지식 기반 및 웹 검색 엔진 기반의 고객 서비스 에이전트 두 개, 쇼핑 지원 에이전트, 판매 에이전트 등

cf)고객 서비스 에이전트와 판매 에이전트

픽션 스튜디오 시스템의 시스템 예시

인간 사용자가 언어 에이전트와 토론에 참여 예시

멀티 에이전트 시스템

소셜 스튜디오, 토론회, 소프트웨어 회사 등

사용 예시

Agents

Agents is an open-source library/framework for building autonomous language agents. If you want to know more about Agents, please check our [Paper](#) and [Github](#). Here is a demo of Agents.

If an error occurs or the queue is too long, please create your own demo by clicking Duplicate This Space in the upper right corner. Please be patient with building, thank you! It takes about 3-4 minutes.

Please input valid openai key for gpt-3.5-turbo-16k.

sk-P5ynTyEPiyJ5szz8RDWYT3BibKfJaSvLHLRYTzESmOFbW3T

Select the execution mode

Single mode refers to when the current agent output ends, it will stop running until you click to continue. Auto mode refers to when you complete the input, all agents will continue to output until the task ends.

☒ Auto Mode ☐ Single Mode

Debate Topic:


Should AI Replace Humans in Creative Fields??

Affirmative viewpoint:

AI should replace humans in creative fields because it can produce art and content efficiently, reduce costs, and eliminate human bias.

Dialog

Affirmative_Task_Allocation_state

John(Affirmative_Debate_organizer)
The debate topic is as follows:

Should AI Replace Humans in Creative Fields?? Affirmative viewpoint: AI should replace humans in creative fields because it can produce art and content efficiently, reduce costs, and eliminate human bias. negative viewpoint: AI should not replace humans in creative fields as it lacks true creativity, emotions, and the ability to understand complex human experiences.

, now , begin to discuss!