언어 에이전트를 위한 인지 아키텍처

Theodore R. Sumers* Shunyu Yao* Karthik Narasimhan Thomas L. Griffiths
Princeton University
{sumers, shunyuy, karthikn, tomg}@princeton.edu

OpenReview에서 검토됨: *https://openreview.net/forum?id=1i6ZCvflQJ*

추상적인

최근의 노력은 접지 또는 추론이 필요한 작업을 위해 외부 리소스(예: 인터넷) 또는 내부 제어 흐름(예: 프롬프트 체이닝)을 사용하여 대규모 언어 모델(LLM)을 증강하여 새로운 클래스의 *language agents*를 만들어냈습니다. 이러한 에이전트는 상당한 경험적 성공을 거두었지만 기존 에이전트를 구성하고 향후 개발을 계획할 프레임워크가 부족합니다. 이 논문에서는 인지 과학과 상징적 인공 지능의 풍부한 역사를 활용하여 언어 에이전트를 위한 인지 아키텍처(CoALA)를 제안합니다. CoALA는 모듈형 메모리 구성 요소, 내부 메모리 및 외부 환경과 상호 작용하는 구조화된 작업 공간, 작업을 선택하기 위한 일반화된 의사 결정 프로세스가 있는 언어 에이전트를 설명합니다. CoALA를 사용하여 최근의 많은 작업을 *retrospectively* 조사하고 구성하고, 보다 유능한 에이전트에 대한 실행 가능한 방향을 *prospectively* 식별합니다. 이를 종합하면 CoALA는 오늘날의 언어 에이전트를 AI의 광범위한 역사 내에서 맥락화하고 언어 기반 일반 지능으로 가는 길을 설명합니다.

1 서론

Language agents (Weng, 2023; Wang et al., 2023b; Xi et al., 2023; Yao and Narasimhan, 2023)은 대규모 언어 모델(LLM, Vaswani et al., 2017; Brown et al., 2020; Devlin et al., 2019; OpenAI, 2023a)을 사용하여 세상과 상호 작용하는 새로운 종류의 인공 지능(AI) 시스템입니다. 그들은 LLM의 최신 발전을 기존 에이전트 설계 분야에 적용합니다(Russell and Norvig, 2013). 흥미롭게도, 이 합성은 두 분야 모두에 이점을 제공합니다. 한편, LLM은 제한된 지식과 추론 능력을 가지고 있습니다. 언어 에이전트는 LLM을 내부 메모리와 환경에 연결하여 기존 지식이나 외부 관찰에 근거함으로써 이러한 문제를 완화합니다. 반면, 전통적인 에이전트는 종종 수작업 규칙(Wilkins, 2014)이나 강화 학습(Sutton and Barto, 2018)이 필요하여 새로운 환경에 대한 일반화가 어렵습니다(Lake et al., 2016). 언어 에이전트는 LLM에 존재하는 상식적 사전 지식을 활용하여 새로운 작업에 적응하여 인간 주석이나 시행착오 학습에 대한 의존도를 줄입니다.

가장 초기의 에이전트는 LLM을 사용하여 행동을 직접 선택하거나 생성했지만(그림 1B; Ahn et al., 2022; Huang et al., 2022b), 최근의 에이전트는 추론(Yao et al., 2022b), 계획(Hao et al., 2023; Yao et al., 2023) 및 장기 기억 관리(Park et al., 2023; Wang et al., 2023a)에도 사용하여 의사 결정을 개선합니다. 이 최신 세대의 *cognitive* 언어 에이전트는 놀라울 정도로 정교한 내부 프로세스를 사용합니다(그림 1C). 그러나 오늘날 개별 작업에서는 이러한 프로세스를 설명하기 위해 사용자 지정 용어(예: '도구 사용', '접지', '행동')를 사용하여 다양한 에이전트를 비교하거나 시간이 지남에 따라 어떻게 진화하는지 이해하거나 깔끔하고 일관된 추상화로 새로운 에이전트를 구축하는 것이 어렵습니다.

이러한 노력을 조직하는 개념적 프레임워크를 확립하기 위해 컴퓨팅과 인공 지능(AI)의 역사에서 두 가지 아이디어인 *production systems*과 *cognitive architectures*를 비교합니다. 생산 시스템은 규칙을 반복적으로 적용하여 일련의 결과를 생성합니다(Newell 및 Simon, 1972). 이는 문자열 조작 시스템(LLM이 해결하는 문제의 유추)에서 시작되었으며 이후 AI 커뮤니티에서 복잡하고 계층적으로 구조화된 시스템을 정의하는 데 채택되었습니다.

*Equal contribution, order decided by coin flip. Each person reserves the right to list their name first. A CoALA-based repo of recent work on language agents: `https://github.com/ysymyth/awesome-language-agents`.

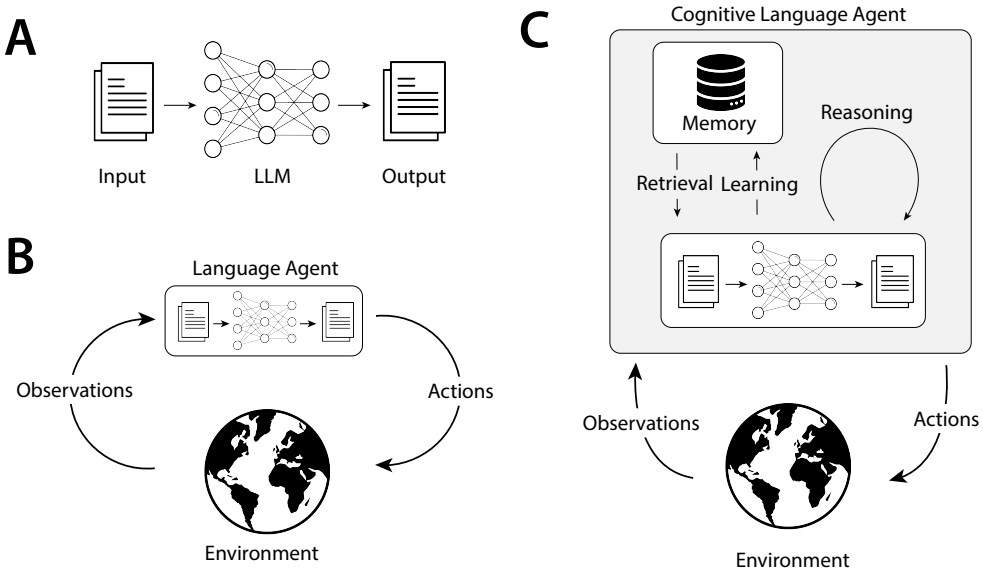

그림 1: 대규모 언어 모델(LLM)의 다양한 용도. A: 자연어 처리(NLP)에서 LLM은 텍스트를 입력으로 받아서 텍스트를 출력합니다. B: *Language agents* (Ahn et al., 2022; Huang et al., 2022c)는 관찰 결과를 텍스트로 변환하고 LLM을 사용하여 작업을 선택함으로써 LLM을 외부 환경과 직접 피드백 루프에 배치합니다. C: *Cognitive* 언어 에이전트(Yao et al., 2022b; Shinn et al., 2023; Wang et al., 2023a)는 또한 LLM을 사용하여 학습 및 추론과 같은 프로세스를 통해 에이전트의 내부 상태를 관리합니다. 이 연구에서 우리는 이러한 에이전트를 구조화하기 위한 청사진을 제안합니다.

행동(Newell et al., 1989). 그렇게 하기 위해, 그들은 새로운 생산을 선택, 적용, 심지어 생성하기 위한 제어 흐름을 지정하는 인지적 아키텍처에 통합되었습니다(Laird et al., 1987; Laird, 2022; Kotseruba and Tsotsos, 2020). 우리는 생산 시스템과 LLM 사이에 의미 있는 유추를 제안합니다. 생산이 문자열을 수정하는 가능한 방법을 나타내는 것처럼, LLM은 텍스트에 대한 변경 또는 추가에 대한 분포를 정의합니다. 이는 생산 시스템과 함께 사용되는 인지적 아키텍처의 제어가 LLM을 언어 에이전트로 변환하는 데 동등하게 적용될 수 있음을 추가로 시사합니다.

따라서 우리는 언어 에이전트를 위한 인지 아키텍처(CoALA)를 제안합니다. 이는 범용 언어 에이전트를 특성화하고 설계하기 위한 개념적 프레임워크입니다. CoALA는 에이전트를 세 가지 핵심 차원에 따라 구성합니다. *information storage* (는 작업 및 장기 기억으로 구분됩니다.)는 내부 및 외부 동작으로 구분됩니다. *action space* (는 계획 및 실행이 있는 상호 작용 루프로 구성됩니다. *decision-making procedure* (는 계획 및 실행이 있는 상호 작용 루프로 구성됩니다.) 이 세 가지 개념(기억, 동작 및 의사 결정)을 통해 CoALA가 기존 에이전트의 많은 부분을 깔끔하게 표현하고 새로운 에이전트를 개발하기 위해 탐색되지 않은 방향을 식별할 수 있음을 보여줍니다. 특히, 최근의 여러 논문에서 일반 지능에 대한 개념적 아키텍처를 제안(LeCun, 2022; McClelland et al., 2019)하거나 언어 모델과 에이전트를 경험적으로 조사(Mialon et al., 2023; Weng, 2023; Wang et al., 2023b)하는 반면, 이 논문은 두 가지 요소를 결합합니다. 우리는 이론적 프레임워크를 제안하고 이를 사용하여 다양한 경험적 작업을 구성합니다. 이를 통해 기존 관행에 대한 이론을 기반으로 삼고 단기 및 장기적 미래 작업 방향을 파악할 수 있습니다.

이 논문의 나머지 부분에 대한 계획은 다음과 같습니다. 먼저 프로덕션 시스템과 인지 아키텍처(섹션 2)를 소개하고 LLM과 언어 에이전트의 최근 개발이 이러한 역사적 아이디어를 어떻게 요약하는지 보여줍니다(섹션 3). 이러한 유사점에 동기를 부여받은 섹션 4에서는 CoALA 프레임워크를 소개하고 이를 사용하여 기존 언어 에이전트를 조사합니다. 섹션 5에서는 여러 저명한 에이전트에 대한 심층적인 사례 연구를 제공합니다. 섹션 6에서는 미래 언어 에이전트를 구성하기 위한 실행 가능한 단계를 제안하는 반면, 섹션 7에서는 인지 과학과 AI의 더 광범위한 아크에서 열린 질문을 강조합니다. 마지막으로 섹션 8에서 결론을 내립니다. 응용 에이전트 설계에 관심이 있는 독자는 섹션 4-6을 우선시할 수 있습니다.

2 배경: 문자열에서 심볼릭 AGI까지

먼저 생산 시스템과 인지 아키텍처를 소개하면서 인지 과학과 인공 지능에 대한 역사적 관점을 제공합니다. 논리와 계산 이론(Post, 1943)으로 시작하여 상징적 인공 일반 지능을 구축하려는 시도(Newell et al., 1989)로 끝납니다. 그런 다음 언어 모델과 언어 에이전트를 간략하게 소개합니다. 섹션 3에서는 이러한 아이디어를 연결하여 생산 시스템과 언어 모델 간의 유사점을 도출합니다.

2.1 문자열 조작을 위한 생산 시스템

20세기 전반에 상당한 지적 작업으로 인해 수학(Whitehead와 Russell, 1997)과 계산(Church, 1932; Turing et al., 1936)이 기호 조작으로 축소되었습니다. 생산 시스템은 그러한 형식주의 중 하나입니다. 직관적으로 생산 시스템은 각각 전제 조건과 동작을 지정하는 일련의 규칙으로 구성됩니다. 전제 조건이 충족되면 동작을 수행할 수 있습니다. 이 아이디어는 계산의 한계를 특성화하려는 노력에서 시작되었습니다. Post(1943)는 이러한 용어로 임의의 논리 시스템에 대한 생각을 제안했습니다. 여기서 공식은 문자열로 표현되고 허용된 결론은 생산 규칙(한 문자열이 다른 문자열을 "생성"하는 것)으로 식별됩니다. 이 공식은 이후 더 간단한 문자열 재작성 시스템과 동일하다는 것이 밝혀졌습니다. 이러한 시스템에서 우리는 다음과 같은 형식의 규칙을 지정합니다.

$$X\,Y\,Z \rightarrow X\,W\,Z$$

문자열 XYZ가 문자열 XWZ로 다시 쓰여질 수 있음을 나타냅니다. 문자열 다시 쓰기는 Chomsky의 구문 구조 문법(Chomsky, 1956)의 형태로 형식 언어 이론에서 중요한 역할을 합니다.

2.2 제어 흐름: 문자열에서 알고리즘까지

그 자체로, 생산 시스템은 단순히 시작점에서 생성될 수 있는 문자열 집합을 특성화합니다. 그러나 *control flow*를 부과하여 실행되는 생산을 결정하는 경우 알고리즘을 지정하는 데 사용할 수 있습니다. 예를 들어, 마르코프 알고리즘은 우선순위 순서가 있는 생산 시스템입니다(Markov, 1954). 다음 알고리즘은 스트로크 $|$로 작성된 숫자를 $Q * R$ 형태로 변환하여 나머지 나누기를 구현합니다. 여기서 Q은 5로 나눈 몫이고 R은 나머지입니다.

$$\begin{aligned} *|||||\ &\rightarrow\ |* \\ *\ &\overset{\bullet}{\rightarrow}\ * \\ &\rightarrow\ * \end{aligned}$$

우선순위 순서가 위에서 아래로 실행되는 경우, 왼쪽에서 오른쪽으로 이동할 때 사전 조건과 일치하는 첫 번째 하위 문자열에 프로덕션이 적용됩니다(마지막 프로덕션의 빈 하위 문자열 포함).

$\overset{\bullet}{\rightarrow}$ 규칙을 실행한 후 알고리즘이 중단됨을 나타냅니다. 첫 번째 규칙은 가능한 경우 5를 효과적으로 "빼기"합니다. 두 번째 규칙은 더 이상 뺄셈이 불가능한 경우 종료 조건을 처리합니다. 세 번째 규칙은 빈 하위 문자열 입력 케이스를 처리합니다. 예를 들어, 입력 11이 주어지면 생성 시퀀스 $*||||||||||| \rightarrow |*|||||| \rightarrow ||*|$가 생성됩니다. $\overset{\bullet}{\rightarrow} ||*|$ 이는 2 나머지 1로 해석됩니다. 간단한 생성은 복잡한 동작을 초래할 수 있습니다. 마르코프 알고리즘은 튜링 완전함을 보여줄 수 있습니다.

2.3 인지 아키텍처: 알고리즘에서 에이전트까지

생산 시스템은 인간의 문제 해결을 포착할 수 있는 형식주의를 찾고 있던 Allen Newell에 의해 AI 커뮤니티에서 대중화되었습니다(Newell, 1967; Newell and Simon, 1972). 생산은 문자열 재작성을 넘어 논리적 연산으로 일반화되었습니다. 에이전트의 목표와 세계 상태에 대해 확인할 수 있는 *preconditions*와 전제 조건이 충족되면 취해야 하는 *actions*입니다. Allen Newell과 Herbert Simon은 획기적인 책 *Human Problem Solving* (Newell and Simon, 1972)에서 다음과 같은 예를 들었습니다.

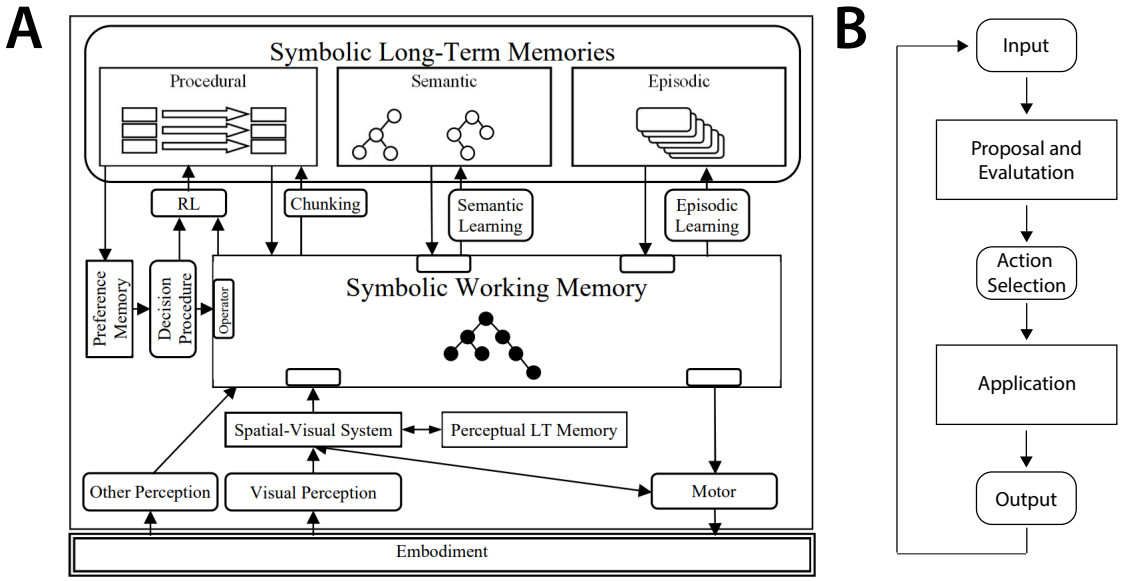

그림 2: 인지 아키텍처는 감각적 근거, 장기 기억, 행동 선택을 위한 결정 절차로 생산 시스템을 증강합니다. A: Soar 아키텍처는 Laird(2022)의 허가를 받아 재생산되었습니다. B: Soar의 결정 절차는 생산을 사용하여 행동을 선택하고 구현합니다. 이러한 행동은 *internal* (에이전트의 기억을 수정하는 것과 같은 것) 또는 *external* (운동 명령과 같은 것)일 수 있습니다.

온도 조절 에이전트를 구현하는 간단한 생산 시스템:

$$
\begin{aligned}
(\text{temperature} > 70^\circ) \wedge (\text{temperature} < 72^\circ) &\rightarrow \text{stop} \\
\text{temperature} < 32^\circ &\rightarrow \text{call for repairs; turn on electric heater} \\
(\text{temperature} < 70^\circ) \wedge (\text{furnace off}) &\rightarrow \text{turn on furnace} \\
(\text{temperature} > 72^\circ) \wedge (\text{furnace on}) &\rightarrow \text{turn off furnace}
\end{aligned}
$$

이 작업 이후, 생산 시스템은 AI 커뮤니티에 의해 채택되었습니다. 그 결과 에이전트는 외부 센서, 액추에이터 및 지식 기반에 연결된 대규모 생산 시스템을 포함했으며, 이에 따라 정교한 제어 흐름이 필요했습니다. AI 연구자들은 인간의 인지를 모방하는 "인지 아키텍처"를 정의했습니다. 이는 지각, 기억 및 계획과 같은 프로세스를 명시적으로 인스턴스화하여(Adams et al., 2012) 유연하고 합리적이며 실시간 행동을 달성했습니다(Sun, 2004; Newell, 1980; 1992; Anderson and Lebiere, 2003). 이는 수백 개의 아키텍처와 수천 개의 출판물을 통해 심리적 모델링에서 로봇 공학에 이르는 응용 프로그램으로 이어졌습니다(최근 조사는 Kotseruba and Tsotsos(2020) 참조).

표준적인 예는 Soar 아키텍처(그림 2A)입니다. Soar는 장기 메모리에 프로덕션을 저장하고 전제 조건이 작업 메모리와 얼마나 잘 일치하는지에 따라 실행합니다(그림 2B). 이러한 프로덕션은 작업 및 장기 메모리의 내용을 수정하는 동작을 지정합니다. 다음으로 Soar에 대한 간략한 개요를 제공하고 독자에게 더 자세한 소개를 위해 Laird(2022; 2019)를 참조하도록 합니다.

기억. 심리학 이론을 바탕으로 Soar는 여러 유형의 기억을 사용하여 에이전트의 상태를 추적합니다(Atkinson 및 Shiffrin, 1968). *Working memory* (Baddeley 및 Hitch, 1974)는 에이전트의 현재 상황을 반영합니다. 에이전트의 최근 지각적 입력, 목표 및 중간 내부 추론의 결과를 저장합니다. *Long term memory*은 세 가지 유형으로 나뉩니다. *Procedural* 기억은 생산 시스템 자체를 저장합니다. 즉, 에이전트의 행동을 결정하기 위해 작업 기억에 적용할 수 있는 규칙 집합입니다. *Semantic* 기억은 세상에 대한 사실을 저장하는 반면(Lindes 및 Laird, 2016), *episodic* 기억은 에이전트의 과거 행동 시퀀스를 저장합니다(Nuxoll 및 Laird, 2007).

접지. Soar는 시뮬레이션(Tambe et al., 1995; Jones et al., 1999) 또는 실제 로봇 시스템(Laird et al., 2012)에서 인스턴스화될 수 있습니다. 구체화된 맥락에서 다양한 센서가 지각 입력을 스트리밍합니다.

작업 기억, 의사 결정에 사용할 수 있는 경우. Soar 에이전트는 액추에이터를 장착하여 물리적 행동과 언어를 통한 상호 작용 학습을 허용할 수도 있습니다(Mohan et al., 2012; Mohan and Laird, 2014; Kirk and Laird, 2014).

의사결정. Soar는 프로덕션을 평가하고 가장 잘 맞는 프로덕션을 적용하는 의사결정 루프를 구현합니다(그림 2B). 프로덕션은 장기 절차 메모리에 저장됩니다. 각 의사결정 주기 동안 해당 전제 조건이 에이전트의 작업 메모리와 비교됩니다. $proposal\ and\ evaluation$ 단계에서 프로덕션 세트를 사용하여 가능한 작업의 후보 세트를 생성하고 순위를 매깁니다.* 그런 다음 가장 좋은 작업을 선택합니다.† 그런 다음 또 다른 프로덕션 세트를 사용하여 작업을 구현합니다. 예를 들어 작업 메모리의 내용을 수정하거나 모터 명령을 내립니다.

학습. Soar는 여러 학습 모드를 지원합니다. 첫째, 새로운 정보는 장기 기억에 직접 저장할 수 있습니다. 사실은 의미 기억에 기록되고 경험은 에피소드 기억에 기록될 수 있습니다(Derbinsky et al., 2012). 이 정보는 나중에 의사 결정에 필요할 때 작업 기억으로 다시 검색할 수 있습니다. 둘째, 행동을 수정할 수 있습니다. 강화 학습(Sutton 및 Barto, 2018)을 사용하여 좋은 결과를 낸 프로덕션을 가중치를 높여 에이전트가 경험으로부터 학습할 수 있습니다(Nason 및 Laird, 2005). 가장 주목할 만한 점은 Soar가 절차적 메모리에 새로운 프로덕션을 기록할 수도 있다는 것입니다(Laird et al., 1986). 효과적으로 소스 코드를 업데이트합니다.

인지 아키텍처는 심리학과 컴퓨터 과학에서 광범위하게 사용되었으며, 로봇공학(Laird et al., 2012), 군사 시뮬레이션(Jones et al., 1999; Tambe et al., 1995), 지능형 튜터링(Koedinger et al., 1997)을 포함한 응용 프로그램이 있습니다. 그러나 지난 수십 년 동안 AI 커뮤니티에서 인기가 떨어졌습니다. 이러한 인기 감소는 이러한 시스템에 관련된 두 가지 과제를 반영합니다. 즉, 논리적 술어로 설명할 수 있는 도메인으로 제한되고 작동하려면 미리 지정된 많은 규칙이 필요합니다.

흥미롭게도 LLM은 이러한 과제를 해결하기에 적합한 것으로 보입니다. 첫째, 임의의 텍스트에서 작동하여 논리 기반 시스템보다 더 유연합니다. 둘째, 사용자가 프로덕션을 지정하도록 요구하는 대신 인터넷 코퍼스에서 사전 학습을 통해 프로덕션에 대한 분포를 학습합니다. 이를 인식한 연구자들은 인지 아키텍처 내에서 LLM을 사용하여 암묵적 세계 지식(Wray et al., 2021)을 활용하여 기존의 상징적 접근 방식(Kirk et al., 2023; Romero et al., 2023)을 증강하기 시작했습니다. 여기서는 대신 인지 아키텍처의 원리를 가져와 LLM 기반 에이전트의 설계를 안내합니다.

2.4 언어 모델 및 에이전트

언어 모델링은 NLP 및 AI 커뮤니티에서 수십 년 된 노력으로, 일부 맥락이 주어지면 텍스트를 생성할 수 있는 시스템을 개발하는 것을 목표로 합니다(Jurafsky, 2000). 형식적으로 언어 모델은 분포 $P(w_i|w_{<i})$를 학습하는데, 여기서 각 w는 개별 토큰(단어)입니다. 그런 다음 이 모델은 한 번에 하나의 토큰씩 분포에서 샘플링하여 텍스트를 생성할 수 있습니다. 언어 모델의 핵심은 확률적 입출력 시스템인데, 텍스트를 계속할 수 있는 방법이 본질적으로 여러 가지 있기 때문입니다(예: "I went to the" $\rightarrow$ "market" | "beach" | ...). 언어(예: n-gram) 모델링에 대한 초기 시도는 일반화 및 확장에서 어려움에 직면했지만, 많은 수(수십억 개)의 매개변수(예: GPT-4; OpenAI, 2023a)와 스마트 토큰화 체계를 갖춘 Transformer 기반(Vaswani et al., 2017) LLM의 등장으로 인해 이 분야가 최근 부활했습니다. 최신 LLM은 엄청난 양의 데이터로 학습되어 많은 수의 입출력 조합에서 지식을 축적하고 인간과 유사한 텍스트를 성공적으로 생성하는 데 도움이 됩니다(Andreas, 2022).

예상치 못하게도 인터넷 규모의 텍스트에서 이러한 모델을 훈련시킨 결과 텍스트 생성을 넘어 코드 작성(Li et al., 2022b; Rozière et al., 2023; Li et al., 2023c), 단백질 모델링(Meier et al., 2021), 대화형 환경에서의 작업(Yao et al., 2022b; Nakano et al., 2021) 등 여러 작업에 유용하게 활용할 수 있었습니다. 후자는 LLM을 핵심 계산 단위로 사용하여 추론, 계획 및 행동하는 시스템인 "언어 에이전트"의 등장으로 이어졌으며 로봇공학(Ahn et al., 2022), 제조(Xia et al., 2023), 웹 조작(Yao et al., 2022a; Deng et al., 2023), 퍼즐 풀기(Yao et al., 2023; Hao et al., 2023) 및 대화형 코드 생성(Yang et al., 2023)과 같은 분야에 응용되었습니다. 언어 이해의 조합

*In more detail, Soar divides productions into two types: “operators,” which we refer to as actions, and “rules” which are used to propose, evaluate, and execute operators.

†If no actions are valid, or multiple actions tie, then an *impasse* occurs. Soar creates a subgoal to resolve the impasse, resulting in hierarchical task decomposition. We refer the reader to Laird (2022) for a more detailed discussion.

그리고 의사결정 능력은 이런 에이전트들을 인간과 같은 지능에 더욱 가깝게 만들어 줄 흥미롭고 새로운 방향입니다.

3 언어 모델과 생산 시스템 간의 연결

문자열 처리에서 공통된 기원을 바탕으로, 생산 시스템과 언어 모델 간에는 자연스러운 유추가 있습니다. 우리는 이 유추를 개발한 다음, 프롬핑 방법이 생산 시스템을 기반으로 하는 알고리즘과 에이전트를 재현한다는 것을 보여줍니다. 생산 시스템과 언어 모델 간의 대응 관계는 우리가 인지 아키텍처를 사용하여 언어 에이전트를 구축하는 데 동기를 부여하며, 이는 섹션 4에서 소개합니다.

3.1 확률적 생산 시스템으로서의 언어 모델

원래 인스턴스화에서 프로덕션 시스템은 시작점에서 생성될 수 있는 문자열 세트를 지정하여 이 프로세스를 일련의 문자열 재작성 작업으로 나누었습니다. 언어 모델은 또한 문자열의 가능한 확장 또는 수정 세트, 즉 모델에 제공된 프롬프트를 정의합니다.[‡]

예를 들어, 텍스트를 완성하는 문제를 생성으로 공식화할 수 있습니다. X가 프롬프트이고 Y이 연속이면 이를 생성 $X \rightarrow X\ Y$로 쓸 수 있습니다.[§] 여러 개의 가능한 연속을 허용하고 싶을 수도 있는데, 이 경우 Y_i의 일부 집합에 대해 $X \rightarrow X\ Y_i$이 있습니다. LLM은 이러한 각 완성에 *probability*을 할당합니다. 이 관점에서 볼 때, LLM은 입력 X이 제공될 때 *which productions to select*에 대한 확률 분포를 정의하여 가능한 완성에 대한 $P(Y_i|X)$ 분포를 생성합니다(Dohan et al., 2022). 따라서 LLM은 호출될 때마다 가능한 완성을 샘플링하는 확률적 생성 시스템으로 볼 수 있습니다(예: $X \rightsquigarrow X\ Y$).

이 확률적 형태는 기존 생산 시스템에 비해 장단점을 모두 제공합니다. LLM의 가장 큰 단점은 본질적인 불투명성입니다. 생산 시스템은 개별적이고 인간이 읽을 수 있는 규칙으로 정의되는 반면, LLM은 수십억 개의 해석할 수 없는 매개변수로 구성됩니다. 이러한 불투명성은 확률적 공식화의 본질적인 무작위성과 결합되어 동작을 분석하거나 제어하기 어렵게 만듭니다(Romero et al., 2023; Valmeekam et al., 2022). 그럼에도 불구하고 규모와 사전 학습은 기존 생산 시스템에 비해 엄청난 이점을 제공합니다. 대규모 인터넷 데이터로 사전 학습된 LLM은 문자열 완성에 대해 놀라울 정도로 효과적인 사전 학습을 수행하여 다양한 작업을 바로 해결할 수 있습니다(Huang et al., 2022b).

3.2 제어 흐름으로서의 신속한 엔지니어링

LLM의 가중치는 입력 문자열(프롬프트)에 따라 조건지어진 출력 문자열(완성)에 대한 우선순위를 정의합니다. 결과 분포는 프로덕션의 작업별 우선순위로 해석할 수 있습니다. 즉, 간단한 제어 흐름입니다. 질문 답변과 같은 작업은 입력 문자열(질문)로 직접 공식화할 수 있으며, 완성(가능한 답변)에 대한 조건부 분포를 생성합니다.

few-shot learning(Brown 등, 2020)과 prompt engineering(Wei 등, 2022b; Kojima 등, 2022; Xu 등, 2023c)에 대한 초기 연구에서는 LLM이 입력 문자열을 사전 처리함으로써 고품질 프로덕션에 더욱 편향될 수 있음을 발견했습니다. 이러한 간단한 조작(일반적으로 입력에 추가 텍스트를 연결)은 프로덕션으로 볼 수 있으며, 이는 이러한 방법이 프로덕션 시퀀스를 정의한다는 것을 의미합니다(표 1). 이후 연구에서는 이러한 접근 방식을 동적이고 상황에 맞는 프롬프트로 확장했습니다. 예를 들어 입력과 최대한 관련성이 있는 few-shot 예제를 선택(Liu 등, 2021)하거나 비디오(Zeng 등, 2022) 또는 데이터베이스(Lewis 등, 2020)의 외부 관찰로 템플릿을 채웁니다. 이러한 프롬프트 기술에 대한 조사는 Liu 등(2023d)을 참조하세요.

이후 작업에서는 LLM 자체를 사전 처리 단계로 사용하여 문제의 특정 측면을 전면에 내세우기 위한 타깃 추론을 이끌어냈습니다(Bai et al., 2022; Jin et al., 2022; Ganguli et al., 2023; Madaan et al., 2023; Saunders et al., 2022; Kim et al., 2023; Kirk et al., 2023) 또는 중간 추론 단계를 생성했습니다(Tafjord

[‡]In this work, we focus on autoregressive LLMs which are typically used for language agents. However, bidirectional LLMs such as BERT (Devlin et al., 2019) can be seen in a similar light: they define a distribution over *in-filling* productions.

[§]Alternatively, we can treat the prompt as input and take the output of the LLM as the next state, represented by the production $X \rightarrow Y$ – a more literal form of rewriting.

Prompting Method	Production Sequence
Zero-shot	$Q \overset{\text{LLM}}{\rightsquigarrow} Q\ A$
Few-shot	$Q \longrightarrow Q_1\ A_1\ Q_2\ A_2\ Q \overset{\text{LLM}}{\rightsquigarrow} Q_1\ A_1\ Q_2\ A_2\ Q\ A$
Retrieval Augmented Generation	$Q \xrightarrow{\text{Wiki}} Q\,O \overset{\text{LLM}}{\rightsquigarrow} Q\,O\,A$
Socratic Models	$Q \overset{\text{VLM}}{\rightsquigarrow} Q\,O \overset{\text{LLM}}{\rightsquigarrow} Q\,O\,A$
Self-Critique	$Q \overset{\text{LLM}}{\rightsquigarrow} Q\,A \overset{\text{LLM}}{\rightsquigarrow} Q\,A\,C \overset{\text{LLM}}{\rightsquigarrow} Q\,A\,C\,A$

표 1: 프롬프팅 방법이 완성을 생성하기 전에 입력 문자열을 조작하는 방식을 설명하는 개념적 다이어그램. Q = 질문, A = 답변, O = 관찰, C = 비판, $\rightsquigarrow$은 확률적 생산에서 샘플링을 나타냅니다. 이러한 사전 처리 조작은 시각 언어 모델(VLM) 또는 LLM 자체와 같은 다른 모델을 사용할 수 있으며, 생산으로 볼 수 있습니다. 따라서 프롬프팅 방법은 생산의 *sequence*을 정의합니다.

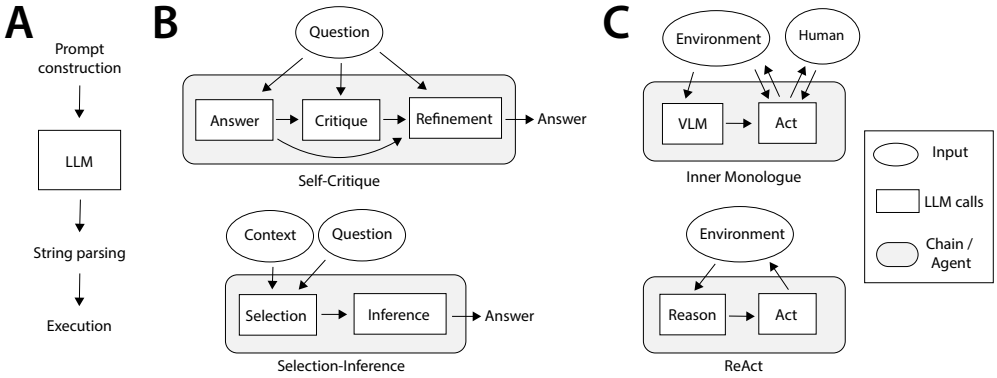

그림 3: 언어 모델에서 언어 에이전트까지. A: LLM 호출의 기본 구조. 프롬프트 구성은 템플릿을 선택하고 작업 메모리의 변수로 채웁니다. LLM을 호출한 후 문자열 출력은 동작 공간으로 구문 분석되어 실행됩니다. LLM 호출은 하나 이상의 동작을 초래할 수 있습니다. 예를 들어, 답변 반환, 함수 호출 또는 모터 명령 발행. B: Self-Critique(Wang et al., 2022b) 또는 Selection-Inference(Creswell et al., 2023)와 같은 *Prompt chaining* 기술은 미리 정의된 LLM 호출 시퀀스를 사용하여 출력을 생성합니다. C: Inner Monologue(Huang et al., 2022c) 및 ReAct(Yao et al., 2022b)와 같은 *Language agents*는 대신 외부 환경과 상호 작용하는 피드백 루프를 사용합니다. 시각 언어 모델(VLM)을 사용하여 지각 데이터를 LLM이 처리할 텍스트로 변환할 수 있습니다.

et al., 2021; Creswell et al., 2023; Yao et al., 2023)이 답변을 반환하기 전에 호출됩니다. *Chaining* LLM에 대한 여러 호출(Wu et al., 2022a;b; Dohan et al., 2022)을 통해 점점 더 복잡한 알고리즘이 가능합니다(그림 3).

3.3 인지 언어 에이전트를 향하여

Language agents 사전 정의된 프롬프트 체인을 넘어서서 대신 LLM을 외부 환경과 피드백 루프에 배치합니다(그림 1B). 이러한 접근 방식은 먼저 다중 모드 입력을 텍스트로 변환하여 LLM에 전달합니다. 그런 다음 LLM의 출력을 구문 분석하여 외부 동작을 결정하는 데 사용합니다(그림 3C). 초기 에이전트는 LLM을 외부 환경과 직접 인터페이스하여 에이전트의 상태에 따라 고수준 지침을 생성하는 데 사용했습니다(Ahn et al., 2022; Huang et al., 2022c; Dasgupta et al., 2022). 이후 작업에서는 동작을 선택하기 전에 LLM을 사용하여 중간 추론을 수행하는 보다 정교한 언어 에이전트가 개발되었습니다(Yao et al., 2022b). 가장 최근의 에이전트는 에피소드 기억을 반영하여 새로운 의미 추론을 생성하거나(Shinn et al., 2023) 자신의

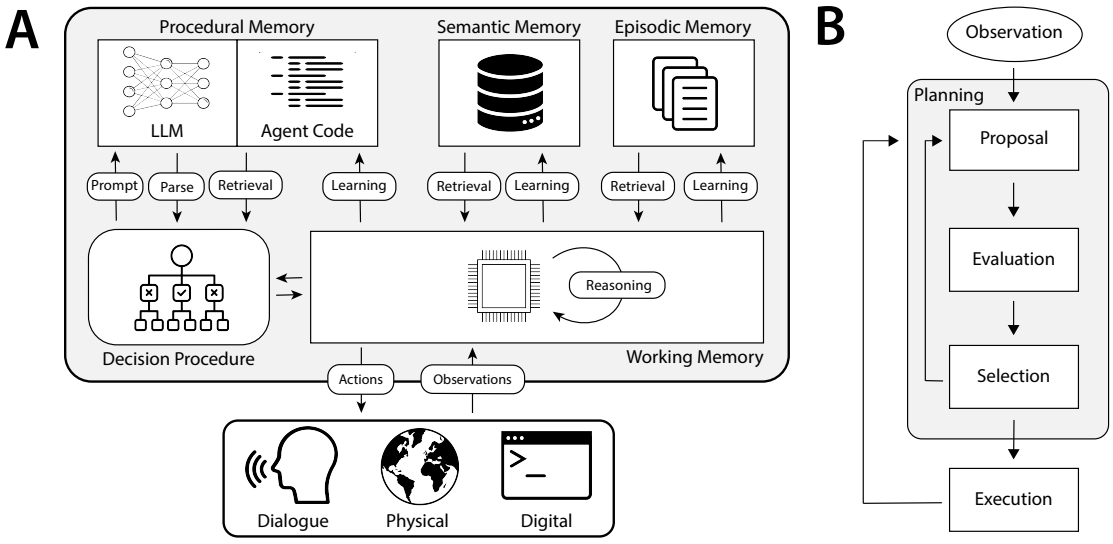

그림 4: 언어 에이전트를 위한 인지 아키텍처(CoALA). A: CoALA는 상호 작용하는 모듈과 프로세스 집합을 정의합니다. 결정 절차는 에이전트의 소스 코드를 실행합니다. 이 소스 코드는 LLM(프롬프트 템플릿 및 파서), 내부 메모리(검색 및 학습), 외부 환경(그라운딩)과 상호 작용하는 절차로 구성됩니다. B: 시간적으로 에이전트의 결정 절차는 외부 환경과 루프에서 결정 주기를 실행합니다. 각 주기 동안 에이전트는 검색 및 추론을 사용하여 후보 학습 또는 그라운딩 작업을 제안하고 평가하여 계획합니다. 그런 다음 최상의 작업을 선택하여 실행합니다. 관찰을 할 수 있으며 주기가 다시 시작됩니다.

절차적 지식을 생성하는 프로그램 코드(Wang et al., 2023a)를 사용하여 이전 경험을 활용하여 미래의 행동을 적응시킵니다.

이 *cognitive* 언어 에이전트는 비사소한 LLM 기반 추론 및 학습을 사용합니다(그림 1C). 인지 아키텍처가 에이전트의 내부 상태 및 외부 환경과 프로덕션 시스템의 상호 작용을 구조화하는 데 사용된 것처럼, 인지 아키텍처가 LLM 기반 인지 에이전트를 설계하는 데 도움이 될 수 있다고 제안합니다. 이 논문의 나머지 부분에서는 이 관점을 사용하여 기존 접근 방식을 구성하고 유망한 확장을 강조합니다.

언어 에이전트를 위한 4가지 인지 아키텍처(CoALA): 개념적 프레임워크

우리는 기존 언어 에이전트를 구성하고 새로운 언어 에이전트의 개발을 안내하는 프레임워크로서 언어 에이전트를 위한 인지 아키텍처(CoALA)를 제시합니다. CoALA는 LLM을 더 큰 인지 아키텍처의 핵심 구성 요소로 위치시킵니다(그림 4). CoALA에서 언어 에이전트는 메모리 모듈에 정보를 저장하고(섹션 4.1) 외부 및 내부 부분으로 구성된 작업 공간에서 작동합니다(그림 5):

- 외부 동작은 접지(섹션 4.2)를 통해 외부 환경과 상호 작용합니다(예: 로봇 제어, 인간과 소통, 웹사이트 탐색).
- 내부 행동은 내부 메모리와 상호 작용합니다. 어떤 메모리에 액세스하는지와 액세스가 읽기인지 쓰기인지에 따라 내부 행동은 세 가지 종류로 더 분해될 수 있습니다. 검색(장기 메모리에서 읽기; 4.3절), 추론(LLM으로 단기 작업 메모리 업데이트; 4.4절), 학습(장기 메모리에 쓰기; 4.5절).

언어 에이전트는 의사 결정을 통해 행동을 선택하는데, 이는 반복적인 사이클을 따릅니다(4.6절, 그림 4B). 각 사이클에서 에이전트는 추론 및 검색 행동을 사용하여 계획을 세울 수 있습니다. 이 계획 하위 프로세스는 외부 세계 또는 에이전트의 장기 기억에 영향을 미치기 위해 실행되는 접지 또는 학습 행동을 선택합니다. CoALA의 의사 결정 사이클은 반환 값이 없는 프로그램의 "main" *procedure* (a *method*와 유사합니다.

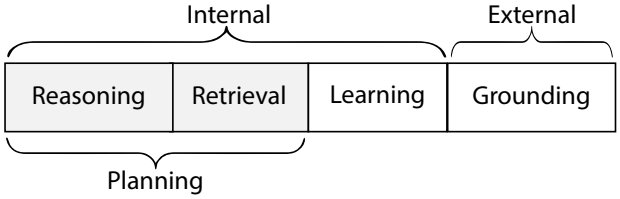

그림 5: 에이전트의 액션 공간은 내부 메모리 액세스와 세계와의 외부 상호작용으로 나눌 수 있습니다. 추론 및 검색 액션은 계획을 지원하는 데 사용됩니다.

*functions*과 대조적으로, 지속적으로 루프를 실행하여 새로운 지각 입력을 허용하고 이에 대한 응답으로 다양한 동작 *procedures*를 호출합니다.

CoALA(그림 4)는 수십 년간의 인지 아키텍처 연구(섹션 2.3)에서 영감을 받아 기억, 근거, 학습 및 의사 결정과 같은 핵심 개념을 활용합니다. 그러나 LLM을 통합하면 "추론" 작업이 추가되어 다양한 목적에 맞게 새로운 지식과 휴리스틱을 유연하게 생성할 수 있으며 기존 인지 아키텍처의 손으로 쓴 규칙을 대체할 수 있습니다. 또한 텍스트를 *de facto* 내부 표현으로 만들어 에이전트의 메모리 모듈을 간소화합니다. 마지막으로 시각 언어 모델(VLM; Alayrac et al., 2022)의 최근 발전은 지각 데이터를 텍스트로 간단하게 변환하여 근거를 단순화할 수 있습니다(Zeng et al., 2022).

이 섹션의 나머지 부분에서는 CoALA의 핵심 개념인 기억, 행동(접지, 추론, 검색 및 학습), 의사 결정을 자세히 설명합니다. 각 개념에 대해 기존 언어 에이전트(또는 관련 NLP/RL 방법)를 예로 사용하거나 향후 방향을 위해 문헌의 간극을 기록합니다.

4.1 메모리

언어 모델은 *stateless*입니다. 호출 간에 정보를 유지하지 않습니다. 반면 언어 에이전트는 세계와의 다단계 상호 작용을 위해 내부적으로 정보를 저장하고 유지할 수 있습니다. CoALA 프레임워크에서 언어 에이전트는 정보(주로 질감이지만 다른 모달리티도 허용됨)를 여러 메모리 모듈로 명시적으로 구성하며, 각각은 다른 형태의 정보를 포함합니다. 여기에는 단기 작업 메모리와 여러 장기 메모리(에피소드, 의미 및 절차)가 포함됩니다.

작업 메모리. 작업 메모리는 현재 의사결정 주기를 위한 상징적 변수로서 활성화되고 즉시 이용 가능한 정보를 유지합니다(섹션 4.6). 여기에는 지각적 입력, 활성 지식(추론에 의해 생성되거나 장기 메모리에서 검색됨) 및 이전 의사결정 주기에서 이월된 기타 핵심 정보(예: 에이전트의 활성 목표)가 포함됩니다. 이전 방법은 LLM이 중간 추론을 생성하도록 장려합니다(Wei et al., 2022b; Nye et al., 2021). 이때 LLM 자체의 컨텍스트를 작업 메모리의 한 형태로 사용합니다. CoALA의 작업 메모리 개념은 더 일반적입니다. 즉, LLM 호출에서 지속되는 데이터 구조입니다. 각 LLM 호출에서 LLM 입력은 작업 메모리의 하위 집합(예: 프롬프트 템플릿 및 관련 변수)에서 합성됩니다. 그런 다음 LLM 출력은 다른 변수(예: 작업 이름 및 인수)로 다시 구문 분석되어 작업 메모리에 다시 저장되고 해당 작업을 실행하는 데 사용됩니다(그림 3A). LLM 외에도 작업 기억은 장기 기억 및 접지 인터페이스와도 상호 작용합니다. 따라서 언어 에이전트의 다양한 구성 요소를 연결하는 중심 허브 역할을 합니다.

에피소드 기억. 에피소드 기억은 이전 의사결정 주기의 경험을 저장합니다. 여기에는 입력-출력 쌍(Rubin et al., 2021), 이력 이벤트 흐름(Weston et al., 2014; Park et al., 2023), 이전 에피소드의 게임 궤적(Yao et al., 2020; Tuyls et al., 2022) 또는 에이전트의 경험에 대한 다른 표현이 포함될 수 있습니다. 의사결정 주기의 계획 단계에서 이러한 에피소드는 추론을 지원하기 위해 작업 메모리로 검색될 수 있습니다. 에이전트는 또한 학습의 한 형태로 작업에서 에피소드 기억으로 새로운 경험을 쓸 수 있습니다(섹션 4.5).

의미 기억. 의미 기억은 세상과 자신에 대한 에이전트의 지식을 저장합니다. 추론이나 의사 결정을 위해 검색을 활용하는 전통적인 NLP 또는 RL 접근 방식은 지식 지원을 위해 외부 데이터베이스에서 의미 기억을 초기화합니다. 예를 들어, NLP의 검색 증강 방법(Lewis et al., 2020; Borgeaud et al., 2022; Chen et al., 2017)은 의미 기억에서 검색하는 것으로 볼 수 있습니다.

비구조화된 텍스트(예: 위키피디아). RL에서 "읽어서 배우기" 접근 방식(Branavan 등, 2012; Narasimhan 등, 2018; Hanjie 등, 2021; Zhong 등, 2021)은 게임 매뉴얼과 사실을 의미 기억으로 활용하여 정책에 영향을 미칩니다. 이러한 예는 기본적으로 고정된 읽기 전용 의미 기억을 사용하지만, 언어 에이전트는 LLM 추론에서 얻은 새로운 지식을 의미 기억에 학습의 한 형태로 기록하여(4.5절) 경험에서 세계 지식을 점진적으로 구축할 수도 있습니다.

절차적 메모리. 언어 에이전트는 두 가지 형태의 절차적 메모리를 포함합니다. $implicit$ LLM 가중치에 저장된 지식과 $explicit$ 에이전트 코드에 작성된 지식입니다. 에이전트의 코드는 두 가지 유형으로 더 나눌 수 있습니다. 추론, 검색, 근거 및 학습 절차와 같은 동작을 구현하는 절차와 의사 결정 자체를 구현하는 절차입니다(4.6절). 의사 결정 주기 동안 LLM은 추론 동작을 통해 액세스할 수 있으며 다양한 코드 기반 절차를 검색하여 실행할 수 있습니다. 처음에는 비어 있거나 없을 수 있는 에피소드적 또는 의미적 메모리와 달리 절차적 메모리는 설계자가 에이전트를 부트스트랩하기 위해 적절한 코드로 초기화해야 합니다. 마지막으로 절차적 메모리에 기록하여 새로운 동작을 학습하는 것은 가능하지만(4.5절), 버그를 쉽게 도입하거나 에이전트가 설계자의 의도를 뒤집을 수 있으므로 에피소드적 또는 의미적 메모리에 기록하는 것보다 상당히 위험합니다.

4.2 접지 동작

접지 절차는 외부 행동을 실행하고 환경 피드백을 텍스트로 작업 메모리에 처리합니다. 이는 텍스트 관찰과 행동을 통해 에이전트가 외부 세계와 상호 작용하는 것을 "텍스트 게임"으로 효과적으로 단순화합니다. 우리는 세 가지 종류의 외부 환경을 분류합니다.

물리적 환경. 물리적 구현은 AI 에이전트에 대해 구상된 가장 오래된 인스턴스화입니다(Nilsson, 1984). 여기에는 지각적 입력(시각적, 청각적, 촉각적)을 텍스트적 관찰(예: 사전 훈련된 캡션 모델을 통해)로 처리하고 언어 기반 명령을 받는 로봇 플래너를 통해 물리적 환경에 영향을 미치는 것이 포함됩니다. LLM의 최근 발전으로 인해 로봇이 물리적 세계에서 작업이나 계획을 생성하기 위해 LLM을 "두뇌"로 활용하는 수많은 로봇 프로젝트(Ahn et al., 2022; Liang et al., 2023a; Singh et al., 2023; Palo et al., 2023; Ren et al., 2023)가 탄생했습니다. 지각 입력의 경우, 일반적으로 시각 언어 모델을 사용하여 이미지를 텍스트로 변환합니다(Alayrac 등, 2022; Sumers 등, 2023). 이는 LLM에 대한 추가적인 맥락을 제공합니다(Driess 등, 2023; Huang 등, 2023; Brohan 등, 2022; 2023).

인간 또는 다른 에이전트와의 대화. 고전적인 언어적 상호 작용을 통해 에이전트는 지시를 수락하거나(Winograd, 1972; Tellex et al., 2011; Chen and Mooney, 2011; Bisk et al., 2016) 사람으로부터 배울 수 있습니다(Nguyen et al., 2021; Sumers et al., 2022; 2021; Wang et al., 2016). $generating$ 언어를 구사할 수 있는 에이전트는 도움(Ren 등, 2023; Nguyen 등, 2022b; 2019; Nguyen 및 Daumé III, 2019)이나 설명(Biyik 및 Palan, 2019; Sadigh 등, 2017; Padmakumar 등, 2022; Thomason 등, 2020; Narayan-Chen 등, 2019)을 요청할 수 있으며, 사람들을 즐겁게 하거나 감정적으로 도울 수 있습니다(Zhang 등, 2020; Zhou 등, 2018; Pataranutaporn 등, 2021; Hasan 등, 2023; Ma 등, 2023). 최근 연구에서는 사회적 시뮬레이션(Park 등, 2023; Jinxin 등, 2023; Gao 등, 2023), 토론(Chan 등, 2023; Liang 등, 2023b; Du 등, 2023), 안전성 향상(Irving 등, 2018), 협력적 과제 해결(Qian 등, 2023; Wu 등, 2023; Hong 등, 2023a; Dong 등, 2023)을 위한 다중 언어 에이전트 간 상호 작용도 조사합니다.

디지털 환경. 여기에는 게임(Hausknecht et al., 2020; Côté et al., 2019; Shridhar et al., 2020; Wang et al., 2022a; Liu et al., 2023e), API(Schick et al., 2023)와의 상호작용이 포함됩니다. ; 야오(Yao) 외, 2022b; 파리시(Parisi) 외, 2022 et al., 2023b) 및 웹사이트(Shi et al., 2017; Nakano et al., 2021; Yao et al., 2022a; Zhou et al., 2023b; Gur et al., 2023; Deng et al., 2023) 뿐만 아니라 일반 코드 실행(Yang et al., 2023; Le et al., 2022; Ni et al., 2023). 이러한 디지털 접지는 물리적 또는 인간 상호 작용보다 저렴하고 빠릅니다. 따라서 언어 에이전트를 위한 편리한 테스트베드이며 최근 몇 년 동안 강도가 점점 더 높아지면서 연구되었습니다. 특히 외부 지식이나 계산의 증강이 필요한 NLP 작업의 경우 , 상태 비저장 디지털 API(예: 검색, 계산기, 번역기)는 종종 "도구"로 패키징됩니다(Parisi et al., 2022; Schick et al., 2023; Xu et al., 2023a; Tang et al., 2023b; Qin et al., 2023)은 특별한 "일회용" 디지털 환경으로 볼 수 있습니다.

4.3 검색 작업

CoALA에서 검색 절차(Li et al., 2022a; Gu et al., 2018)는 장기 기억에서 작업 기억으로 정보를 읽어들입니다. 정보와 기억 유형에 따라 규칙 기반, 희소 또는 밀집 검색 등 다양한 방식으로 구현될 수 있습니다. 예를 들어, Voyager(Wang et al., 2023a)는 밀집 검색을 통해 스킬 라이브러리에서 코드 기반 스킬을 로드하여 Minecraft 세계와 상호 작용합니다. 이는 절차적 기억에서 접지 절차를 효과적으로 검색합니다. Generative Agents(Park et al., 2023)는 최근성(규칙 기반), 중요성(추론 기반), 관련성(임베딩 기반) 점수의 조합을 통해 에피소드 기억에서 관련 이벤트를 검색합니다. DocPrompting(Zhou et al., 2022a)은 라이브러리 문서를 활용하여 코드 생성을 지원하는 것을 제안하는데, 이는 의미적 기억에서 지식을 검색하는 것으로 볼 수 있습니다. 검색은 인간의 의사 결정에서 핵심적인 역할을 하지만(Zhou et al., 2023a; Zhao et al., 2022), 언어 에이전트에서 적응적이고 맥락에 따른 회상은 여전히 충분히 연구되지 않았습니다. 섹션 6에서 우리는 의사 결정과 검색의 원칙적인 통합을 중요한 미래 방향으로 제안합니다.

4.4 추론 작업

추론은 언어 에이전트가 작업 기억의 내용을 처리하여 새로운 정보를 생성할 수 있도록 합니다. 검색(장기 기억에서 작업 기억으로 읽는 것)과 달리 추론은 and에서 읽고 작업 기억에 씁니다. 이를 통해 에이전트는 가장 최근의 관찰(Yao et al., 2022b; Peng et al., 2023), 가장 최근의 궤적(Shinn et al., 2023) 또는 장기 기억에서 검색된 정보(Park et al., 2023)에 대한 통찰력을 요약하고 추출할 수 있습니다. 추론은 학습(결과를 장기 기억에 기록) 또는 의사 결정(결과를 후속 LLM 호출에 대한 추가 맥락으로 사용)을 지원하는 데 사용할 수 있습니다.

4.5 학습 활동

학습은 장기 기억에 정보를 기록하는 방식으로 이루어지는데, 여기에는 다양한 절차가 포함됩니다.

경험으로 에피소드 기억 업데이트. RL 에이전트가 매개변수 정책을 업데이트하기 위해 에피소드 궤적을 저장하는 것은 일반적인 관행입니다(Blundell et al., 2016; Pritzel et al., 2017) 또는 비매개변수 정책을 수립합니다(Ecoffet et al., 2019; Tuyls et al., 2022). 언어 에이전트의 경우 에피소드 기억에 추가된 경험은 나중에 추론이나 의사 결정을 위한 예와 기반으로 검색될 수 있습니다(Weston et al., 2014; Rubin et al., 2021; Park et al., 2023).

지식으로 의미 기억 업데이트. 최근 연구(Shinn et al., 2023; Park et al., 2023)에서는 LLM을 적용하여 원시 경험에 대해 추론하고 그 결과 추론을 의미 기억에 저장합니다. 예를 들어, Reflexion(Shinn et al., 2023)은 LLM을 사용하여 실패한 에피소드를 반성하고 결과(예: "주방에 식기 세척기가 없습니다")를 의미 지식으로 저장하여 이후 에피소드를 해결하기 위해 LLM 컨텍스트에 첨부합니다. 마지막으로, 로봇 공학 연구(Chen et al., 2023a)에서는 시각 언어 모델을 사용하여 환경의 의미 맵을 구축하여 나중에 쿼리하여 명령을 실행할 수 있습니다.

LLM 매개변수(절차적 메모리) 업데이트. LLM 가중치는 암묵적 절차적 지식을 나타냅니다. 이는 에이전트의 수명 동안 미세 조정을 통해 에이전트의 도메인에 맞게 조정할 수 있습니다. 이러한 미세 조정은 지도 학습(Liu 등, 2023c; Zhang 등, 2023b) 또는 모방 학습(Hussein 등, 2017), 환경 피드백을 통한 강화 학습(RL)(Sutton 및 Barto, 2018), 인간 피드백(RLHF; Christiano 등, 2017; Ouyang 등, 2022; Nakano 등, 2021) 또는 AI 피드백(Bai 등, 2022; Liu 등, 2023f)을 통해 달성할 수 있습니다. 고전적 LLM 자기개선 방법(Huang 등, 2022a; Zelikman 등, 2022)은 일관성과 같은 외부 측정을 사용하여 미세 조정할 세대를 선택합니다(Wang 등, 2022b). 강화 학습 설정에서 이를 확장하여 대신 환경 피드백을 사용할 수 있습니다. 예를 들어, XTX(Tuyls 등, 2022)는 에피소드 메모리에 저장된 고득점 궤적에서 작은 언어 모델을 주기적으로 미세 조정하여 확률성에 직면하여 탐색 프런티어에 도달하기 위한 강력한 "활용" 정책 역할을 합니다. 에이전트의 LLM을 미세 조정하는 것은 비용이 많이 드는 학습 형태이므로 현재 연구에서는 학습 일정을 지정합니다. 그러나 훈련이 더 효율적이 되거나 에이전트가 더 작은 하위 작업별 LLM을 활용하는 경우 언어 에이전트가 LLM을 언제 어떻게 미세 조정할지 자율적으로 결정할 수 있도록 할 수 있습니다.

에이전트 코드(절차적 메모리) 업데이트. CoALA는 에이전트가 소스 코드를 업데이트하여 다양한 절차의 구현을 수정할 수 있도록 합니다. 이는 다음과 같이 구분할 수 있습니다.

- 추론 업데이트(예: 프롬프트 템플릿; Gao et al., 2020; Zhou et al., 2022b). 예를 들어, APE(Zhou et al., 2022b)는 입출력 예제에서 프롬프트 지침을 추론한 다음 이러한 지침을 LLM 프롬프트의 일부로 사용하여 작업 해결을 지원합니다. 이러한 프롬프트 업데이트는 추론 학습의 한 형태로 볼 수 있습니다.

- 접지 업데이트(예: 코드 기반 기술; Liang et al., 2023a; Ellis et al., 2021; Wang et al., 2023a). 예를 들어, Voyager(Wang et al., 2023a)는 커리큘럼 라이브러리를 유지 관리합니다. 주목할 점은 현재 방법은 외부 환경과 상호 작용하기 위한 새로운 코드 기술을 만드는 데 국한되어 있다는 것입니다.
- 검색 업데이트. 우리가 아는 한, 이러한 학습 옵션은 최근 언어 에이전트에서 연구되지 않았습니다. 검색은 일반적으로 일부 고정된 구현(예: BM25 또는 밀집 검색)으로 설계된 기본 동작으로 간주되지만 쿼리/문서 확장(Nogueira 등, 2019; Wang 등, 2023c; Tang 등, 2023a) 또는 검색 증류(Izacard 등, 2021)에 대한 연구는 언어 에이전트가 더 나은 검색 절차를 학습하는 데 도움이 될 수 있습니다.

- 학습 또는 의사결정 업데이트. 마지막으로, CoALA 에이전트는 학습 또는 의사결정을 위한 새로운 절차를 학습하여 상당한 적응성을 제공할 수 있습니다. 그러나 일반적으로 이러한 절차에 대한 업데이트는 에이전트의 기능과 정렬에 모두 위험합니다. 현재로서는 이러한 형태의 학습을 구현하는 언어 에이전트를 알지 못합니다. 이러한 가능성에 대해서는 섹션 6에서 더 자세히 설명합니다.

RL 에이전트는 일반적으로 한 가지 학습 방식(예: Q-학습, PPO 또는 A3C)을 고정하고 모델 매개변수를 업데이트하여 학습하는 반면, 언어 에이전트는 다양한 학습 절차 중에서 선택할 수 있습니다. 이를 통해 작업 관련 언어를 저장하여 빠르게 학습하고(매개변수 업데이트보다 저렴하고 빠름) 여러 학습 형태를 활용하여 자체 개선을 강화할 수 있습니다(예: 섹션 5에서 논의한 생성 에이전트).

마지막으로, 우리의 논의는 주로 메모리에 추가하는 것에 집중되었지만, 수정 및 삭제("언러닝"의 경우)는 최근 언어 에이전트에서 충분히 연구되지 않았습니다. 우리는 섹션 6에서 이러한 영역을 더 자세히 다룹니다.

4.6 의사결정

행동 공간에서 다양한 행동(접지, 학습, 추론, 검색)을 통해 언어 에이전트는 어떤 행동을 적용할지 어떻게 선택해야 할까요? 이는 의사 결정 절차에 의해 처리되는데, 이는 사실상 최상위 또는 "주요" 에이전트 프로그램입니다. CoALA는 이 최상위 프로그램을 의사 결정 주기(그림 4B)로 구조화하여 외부 $grounding$ 행동(4.2절) 또는 내부 $learning$ 행동(4.5절)을 생성합니다. 각 주기에서 프로그램 코드는 대안을 제안하고 평가하기 위한 추론 및 검색 행동 시퀀스를 정의한 다음(계획 단계) 선택한 행동을 실행합니다(실행 단계). 그런 다음 주기가 다시 반복됩니다.

계획 단계. 계획하는 동안 추론 및 검색을 유연하게 적용하여 작업을 제안, 평가 및 선택할 수 있으며, 이러한 하위 단계는 외부 작업을 수행하기 전에(Yao et al., 2023; Hao et al., 2023) 여러 단계의 시뮬레이션을 구축하기 위해 교차하거나 반복할 수 있습니다. 또한 에이전트가 후보 솔루션을 반복적으로 개선할 수 있도록 합니다. 예를 들어 LLM을 사용하여 시뮬레이션하고, 결함을 식별하고, 해당 결함을 해결하는 수정을 제안합니다(Kirk et al., 2023; Shinn et al., 2023).

- 제안. 제안 하위 단계는 하나 이상의 작업 후보를 생성합니다. 일반적인 접근 방식은 추론(및 선택적으로 검색)을 사용하여 LLM에서 하나(Huang 등, 2022c) 또는 그 이상(Chen 등, 2021; Wang 등, 2022b)의 외부 접지 작업을 샘플링하는 것입니다. 작업이 제한된 간단한 도메인의 경우 제안 단계에 모든 작업(예: 섹션 5의 SayCan)을 포함할 수 있습니다. 보다 정교한 에이전트는 if-else 또는 while-if 코드 구조를 사용합니다(Wang 등, 2023a; Park 등, 2023). 반면 잘 정의된 도메인에 배포된 에이전트는 구조화된 시뮬레이터(Haslum 등, 2019)를 활용하여 그럴듯한 롤아웃(Liu 등, 2023a; Dagan 등, 2023)을 생성할 수 있습니다.
- 평가. 여러 가지 조치가 제안된 경우 평가 하위 단계에서는 각각에 값을 할당합니다. 여기에는 휴리스틱 규칙, LLM(퍼플렉시티) 값(Ahn et al., 2022), 학습된 값(Yao et al.,

	Long-term Memory[¶]	External Grounding	Internal Actions	Decision Making
SayCan (Ahn et al., 2022)	-	physical	-	evaluate
ReAct (Yao et al., 2022b)	-	digital	reason	propose
Voyager (Wang et al., 2023a)	procedural	digital	reason/retrieve/learn	propose
Generative Agents (Park et al., 2023)	episodic/semantic	digital/agent	reason/retrieve/learn	propose
Tree of Thoughts (Yao et al., 2023)	-	digital[‖]	reason	propose, evaluate, select

표 2: CoALA 프레임워크에 적용된 최근 언어 에이전트 중 일부.

2020), LLM 추론(Yao et al., 2023; Hao et al., 2023) 또는 이들의 조합. 특히, LLM 추론은 외부 세계로부터의 접지 피드백을 내부적으로 시뮬레이션하여 행동을 평가하는 데 도움이 될 수 있습니다(Hao et al., 2023; Yang et al., 2023).

- 선택. 일련의 동작과 그 값이 주어지면 선택 단계는 실행할 동작을 선택하거나 거부하고 제안 단계로 돌아갑니다. 동작 값의 형태에 따라 선택은 argmax, softmax 또는 다수결 투표와 같은 대안을 통해 발생할 수 있습니다(Wang et al., 2022b).

실행. 선택된 작업은 에이전트의 소스 코드에서 관련 프로시저를 실행하여 적용됩니다. 에이전트 구현에 따라 이는 외부 *grounding* 작업(예: API 호출, 섹션 4.2)이거나 내부 *learning* 작업(예: 에피소드 메모리에 쓰기, 섹션 4.5)일 수 있습니다. 환경에서 관찰하여 에이전트의 작업에서 피드백을 제공하고 사이클이 다시 반복됩니다.

경험적으로, 많은 초기 언어 에이전트는 중간 추론이나 검색 없이 단순히 LLM을 사용하여 동작(Schick 등, 2023), 동작 시퀀스(Huang 등, 2022b)를 제안하거나 고정된 동작 집합을 평가(Ahn 등, 2022)합니다. 후속 작업(Yao 등, 2022b; Shinn 등, 2023; Xu 등, 2023b; Lin 등, 2023; Wang 등, 2023a; Park 등, 2023)에서는 중간 추론 및 검색을 활용하여 상황을 분석하고, 동작 계획을 수립하고 유지하고, 환경 피드백을 고려하여 이전 동작을 개선하고, 더 복잡한 절차를 활용하여 단일 동작을 제안했습니다. 가장 최근에는 반복적인 제안 및 평가를 사용하여 여러 동작을 고려하는 더 복잡한 의사 결정을 조사하기 위한 연구가 시작되었습니다. 이러한 절차는 고전적인 계획 알고리즘을 모델로 합니다. 예를 들어, Tree of Thoughts(Yao et al., 2023)와 RAP(Hao et al., 2023)는 LLM을 사용하여 각각 BFS/DFS와 Monte Carlo Tree Search(MCTS; Browne et al., 2012)를 구현합니다. LLM은 제안을 생성(즉, 작업에 따라 롤아웃을 시뮬레이션)하고 이를 평가(즉, 제안된 작업의 결과를 평가)하는 데 사용됩니다.

5가지 사례 연구

CoALA는 메모리 모듈, 액션 공간, 의사 결정 절차의 변형과 절제를 통해 광범위한 언어 에이전트를 표현할 수 있습니다. 표 2는 Minecraft에서 로봇 공학, 순수 추론에서 사회적 시뮬라크라에 이르기까지 다양한 도메인에서 인기 있는 최근 방법을 나열합니다. CoALA는 내부 메커니즘을 특성화하고 간단하고 체계적인 방식으로 유사점과 차이점을 드러내는 데 도움이 됩니다.

SayCan(Ahn et al., 2022)은 사용자 명령을 충족하기 위해 주방에서 로봇 상호 작용에 언어 모델을 근거로 합니다(예: "방금 운동했는데, 회복할 음료와 간식을 가져다 줄래?"). 장기 기억은 절차적일 뿐입니다(LLM과 학습된 가치 함수). 행동 공간은 외부적일 뿐입니다. 즉, 551개의 근거 기술(예: "사과 찾기", "테이블로 가기")의 고정된 집합이며 추론, 검색 또는 학습의 내부 행동은 없습니다. 의사 결정 중에 SayCan은 LLM과 학습된 값을 조합하여 각 행동을 평가하여 기술의 유용성과 근거성을 균형 있게 조절합니다. 따라서 SayCan은 LLM(학습된 가치 함수와 함께)을 단일 단계 플래너로 사용합니다.

ReAct(Yao et al., 2022b)는 다양한 디지털 환경(예: 위키피디아 API, 텍스트 게임, 웹사이트)에 기반을 둔 언어 에이전트입니다. SayCan과 마찬가지로 의미적 또는 에피소드적 기억이 없기 때문에 검색 또는 학습 동작이 없습니다. 동작 공간은 (내부) 추론과 (외부) 근거로 구성됩니다. 의사 결정 주기는 단일 추론 동작을 사용하여 상황을 분석하고 동작 계획을 (다시) 만든 다음

[¶]All agents contain some procedural memory (agent code and LLM weights), so here we only list writable procedural memory.
[‖]Special digital grounding with the only external action being submitting a final answer.

평가나 선택 단계 없이 접지 액션을 수행합니다. ReAct는 내부 및 외부 액션을 모두 활용하는 가장 간단한 언어 에이전트로 간주될 수 있으며, 이러한 액션의 시너지 효과를 보여주는 최초의 작업입니다. 추론은 행동을 안내하는 데 도움이 되고, 행동은 추론을 지원하기 위한 환경적 피드백을 제공합니다.

Voyager(Wang et al., 2023a)는 Minecraft API에 기반을 둔 언어 에이전트입니다. 학습된 가치 함수를 통해 지각에 기반을 두는 SayCan과 달리 Voyager의 기반은 텍스트 전용입니다. Voyager는 코드 기반 기반 절차 라이브러리, 즉 기술(예: "combatZombie", "craftStoneSword")을 저장하는 장기 절차 메모리를 가지고 있습니다. 이 라이브러리는 계층적입니다. 복잡한 기술은 더 간단한 기술을 하위 절차로 사용할 수 있습니다(예: "combatZombie"는 인벤토리에 검이 없으면 "craftStoneSword"를 호출할 수 있음). 가장 인상적인 점은 작업 공간에 기반, 추론, 검색 및 학습(새로운 기반 절차 추가)의 네 가지 유형의 작업이 모두 있다는 것입니다. 의사 결정 주기 동안 Voyager는 작업 메모리에 누락된 경우 먼저 새 작업 목표를 제안하는 이유를 제시한 다음, 작업을 해결하기 위한 코드 기반 기반 기반 절차를 제안하는 이유를 제시합니다. 다음 결정 주기에서 Voyager는 환경 피드백을 추론하여 작업 완료를 결정합니다. 성공하면 Voyager는 절차적 메모리에 접지 절차를 추가하는 학습 작업을 선택합니다. 그렇지 않으면 추론을 사용하여 코드를 정제하고 다시 실행합니다. 장기 메모리와 절차적 학습의 중요성은 ReAct 및 AutoGPT와 같은 기준선과 절차적 메모리가 없는 절제와 비교하여 경험적으로 검증됩니다. Voyager는 영역을 더 잘 탐색하고, 기술 트리를 마스터하고, 보이지 않는 작업에 대한 제로 샷 일반화를 수행하는 것으로 나타났습니다.

생성 에이전트(Park et al., 2023)는 샌드박스 게임에 기반을 둔 언어 에이전트로, 환경 및 다른 에이전트와 상호 작용할 수 있습니다. 또한 액션 공간에는 접지, 추론, 검색 및 학습의 네 가지 액션이 모두 있습니다. 각 에이전트에는 이벤트를 목록으로 저장하는 장기 에피소드 메모리가 있습니다. 이러한 에이전트는 검색 및 추론을 사용하여 에피소드 메모리에 대한 반성(예: "지금 스키를 타는 것을 좋아해요.")을 생성한 다음 장기 의미 메모리에 기록합니다. 의사 결정 중에 의미 메모리에서 관련 반성을 검색한 다음 하루의 고수준 계획을 세우는 이유를 설명합니다. 계획을 실행하는 동안 에이전트는 접지 관찰 스트림을 받습니다. 이를 추론하여 계획을 유지하거나 조정할 수 있습니다.

Tree of Thoughts(ToT)(Yao et al., 2023)는 단 하나의 외부 행동, 즉 추론 문제(24게임, 창의적 글쓰기, 크로스워드 퍼즐)에 대한 최종 해결책을 제출하는 특별한 종류의 언어 에이전트로 볼 수 있습니다. 장기 기억이 없고 내부 행동 공간에서만 추론하지만 의도적인 의사 결정에서 이전의 모든 에이전트와 다릅니다. 계획하는 동안 ToT는 LLM 추론을 기반으로 "생각"(추론 행동)을 반복적으로 제안, 평가 및 선택하고 트리 검색 알고리즘을 통해 이를 유지하여 전역 탐색과 지역적 백트랙 및 예측을 가능하게 합니다.

6가지 실행 가능한 통찰력

언어 에이전트를 중심으로 한 최근의 경험적 조사(Mialon et al., 2023; Weng, 2023; Wang et al., 2023b)와 비교했을 때, CoALA는 인지 아키텍처에 대한 잘 확립된 연구에 기반한 이론적 프레임워크를 제공합니다. 이는 독특하고 보완적인 실행 가능한 통찰력 세트로 이어집니다.

모듈형 에이전트: 모놀리스를 넘어서 생각하기. 아마도 우리의 가장 중요한 제안은 $agents\ should\ be\ structured\ and\ modular$일 것입니다. 실제로 표준화된 소프트웨어가 로봇 플랫폼 전반에 사용되는 것처럼(Quigley, 2009; Macenski et al., 2022), 언어 에이전트를 위한 프레임워크는 기술 투자를 통합하고 호환성을 개선할 것입니다.

- 학술 연구에서 표준화된 용어는 작업 간의 개념적 비교를 허용하고(표 2), 오픈 소스 구현은 모듈형 플러그 앤 플레이 및 재사용을 더욱 용이하게 합니다. 예를 들어, 마르코프 결정 프로세스(Puterman, 2014)의 이론적 프레임워크는 강화 학습(Sutton 및 Barto, 2018)을 위한 표준화된 개념 및 용어(예: 상태, 동작, 보상, 전환) 세트를 제공합니다. 이에 따라 OpenAI Gym(Brockman et al., 2016)과 같은 경험적 프레임워크는 경험적 RL 작업을 용이하게 하는 표준화된 추상화(예: `obs, reward, done, info = env.step(action)`)를 제공합니다. 따라서 언어 에이전트에 대한 유용한 추상화(예: `Memory`, `Action`, `Agent` 클래스)를 구현하고 더 간단한 에이전트를 이러한 경험적 CoALA 프레임워크에 캐스팅하여 더 복잡한 에이전트를 구축하기 위한 예로 사용하는 것이 시기적절하고 효과적일 것입니다.

- 산업 응용 프로그램에서 단일 회사 전체 "언어 에이전트 라이브러리"를 유지하면 개별 에이전트 배포에서 테스트 및 구성 요소 재사용을 용이하게 하여 기술 부채를 줄일 수 있습니다(Sculley et al., 2014; Lwakatare et al., 2020). 또한 고객 경험을 표준화할 수도 있습니다. 개별 팀이 개발한 언어 에이전트의 혼합물과 상호 작용하는 대신 최종 사용자는 동일한 기본 에이전트의 컨텍스트별 인스턴스화를 경험하게 됩니다.
- LLM 대 에이전트 설계 코드. CoALA 에이전트는 두 가지 형태의 절차적 메모리를 가지고 있습니다. 에이전트 코드(결정적 규칙)와 LLM 매개변수(대규모 확률적 생산 시스템)입니다. 에이전트 코드는 해석 가능하고 확장 가능하지만 확률성에 직면하여 종종 취약하고 설계자가 예상하는 상황을 해결하는 데 제한됩니다. 반면 LLM 매개변수는 해석하기 어렵지만 새로운 맥락에서 상당한 제로샷 유연성을 제공합니다(Huang et al., 2022b). 따라서 CoALA는 LLM 제한을 보완하는 일반 알고리즘을 구현하기 위해 코드를 아껴서 사용할 것을 제안합니다. 예를 들어, 자기 회귀 생성으로 인한 근시를 완화하기 위해 트리 검색을 구현합니다(Yao et al., 2023; Hao et al., 2023).

에이전트 설계: 단순한 추론을 넘어서는 사고. CoALA는 세 가지 뚜렷한 개념에 따라 에이전트를 정의합니다. (i) 내부 메모리, (ii) 가능한 내부 및 외부 작업 집합, (iii) 해당 작업에 대한 의사 결정 절차. CoALA를 사용하여 애플리케이션별 에이전트를 개발하려면 이러한 각 구성 요소에 대한 구현을 차례로 지정합니다. 에이전트의 환경과 외부 작업 공간이 주어진다고 가정하고 CoALA를 사용하여 적절한 고수준 아키텍처를 결정하는 방법을 보여줍니다. 예를 들어, 사용자가 쿼리와 구매 내역을 기반으로 관련 항목을 찾는 데 도움이 되는 개인화된 리테일 어시스턴트(Yao et al., 2022a)를 설계한다고 상상해 보세요. 이 경우 외부 작업은 대화 또는 사용자에게 검색 결과를 반환하는 것으로 구성됩니다.

- 어떤 메모리 모듈이 필요한지 확인합니다. 우리의 리테일 어시스턴트 사례에서 에이전트는 판매할 품목 세트를 포함하는 의미적 메모리와 각 고객의 이전 구매 및 상호 작용에 대한 에피소드적 메모리를 갖는 것이 도움이 될 것입니다. 이러한 데이터 저장소를 쿼리하는 함수를 정의하는 절차적 메모리와 대화 상태를 추적하는 작업 메모리가 필요합니다.
- 에이전트의 내부 액션 공간을 정의합니다. 이는 주로 에이전트의 각 메모리 모듈에 대한 읽기 및 쓰기 액세스를 정의하는 것으로 구성됩니다. 우리의 예에서 에이전트는 에피소드 메모리에 대한 읽기 및 쓰기 액세스 권한이 있어야 하지만(따라서 고객과의 새로운 상호 작용을 저장할 수 있음), 의미 및 절차 메모리에 대한 읽기 전용 액세스 권한이 있어야 합니다(인벤토리나 자체 코드를 업데이트해서는 안 됨).
- 의사 결정 절차를 정의합니다. 이 단계는 외부 또는 학습 작업을 선택하기 위해 추론 및 검색 작업이 수행되는 방식을 지정합니다. 일반적으로 이는 성능과 일반화 간의 균형을 필요로 합니다. 더 복잡한 절차는 특정 문제에 더 잘 맞을 수 있습니다(예: Minecraft의 경우 Voyager(Wang et al., 2023a)). 반면 더 간단한 절차는 도메인에 더 독립적이며 일반화할 수 있습니다(예: ReAct(Yao et al., 2022b)). 리테일 어시스턴트의 경우 사용자와의 상호 작용에 대한 에피소드 메모리를 검색하여 검색 의도에 대한 사전 정보를 제공하고 특정 검색 결과 집합이 해당 의도를 충족하는지 여부를 추론하는 명시적 평가 단계를 장려할 수 있습니다. 상호 작용의 끝까지 학습을 연기하여(Shinn et al., 2023; Park et al., 2023) 에피소드를 에피소드 메모리에 저장하기 전에 요약함으로써 의사 결정 절차를 간소화할 수 있습니다.

구조화된 추론: 프롬프트 엔지니어링을 넘어서는 사고. 프롬프트 엔지니어링에 대한 초기 작업은 저수준 문자열 연산을 통해 LLM의 입력과 출력을 조작했습니다. CoALA는 작업 메모리 변수를 업데이트하기 위해 보다 구조화된 추론 절차를 제안합니다.

- LangChain(LangChain, 2022) 및 LlamaIndex(LlamaIndex, 2023)와 같은 프롬프팅 프레임워크를 사용하면 더 높은 수준의 추론 단계 시퀀스를 정의하여 LLM 호출당 추론 부담과 저수준 프롬프트 제작 노력을 줄일 수 있습니다. Guidance(Guidance, 2023) 및 OpenAI 함수 호출(OpenAI, 2023b)과 같은 구조적 출력 구문 분석 솔루션은 작업 메모리 변수를 업데이트하는 데 도움이 될 수 있습니다. 우수한 작업 메모리 모듈을 정의하고 구축하는 것도 미래 연구의 중요한 방향이 될 것입니다. 이러한 모듈은 LLM 추론이 대규모 코드 인프라와 원활하게 통합되어야 하는 산업 솔루션에 특히 중요할 수 있습니다.

- 에이전트의 추론 사용 사례는 유형(예: 자기 평가, 반성, 행동 생성 등을 위한 추론) 및 형식(예: CoT(Wei 등, 2022b), ReAct(Yao 등, 2022b), Reflexion(Shinn 등, 2023)) 측면에서 LLM 교육을 알리고 재구성할 수 있습니다. 기본적으로 기존 LLM은 NLP 작업에 대해 교육되고 최적화되지만 에이전트 애플리케이션은 광범위하게 유용한 것으로 입증된 새로운 LLM 추론 모드(예: 자기 평가)를 탐색했습니다. 이러한 기능에 대해 교육되거나 미세 조정된 LLM은 미래 에이전트의 중추가 될 가능성이 더 큽니다.

장기 기억: 검색 증강을 넘어서 생각하기. 기존의 검색 증강 언어 모델(Guu et al., 2020; Lewis et al., 2020; Borgeaud et al., 2022)은 인간이 쓴 코퍼스에서만 읽을 수 있는 반면, 기억 증강 언어 에이전트는 자체 생성 콘텐츠를 자율적으로 읽고 쓸 수 있습니다. 이를 통해 효율적인 평생 학습을 위한 수많은 가능성이 열립니다.

- 기존 인간 지식과 새로운 경험 및 기술을 결합하면 에이전트가 효율적으로 부트스트랩하여 학습하는 데 도움이 될 수 있습니다. 예를 들어, 코드 작성 에이전트는 매뉴얼이나 교과서의 형태로 의미론적 프로그래밍 지식을 부여받을 수 있습니다. 그런 다음 경험에서 자체 에피소드 지식을 생성할 수 있습니다. 이러한 경험을 반영하여 새로운 의미론적 지식을 생성하고 유용한 방법을 저장하는 코드 라이브러리의 형태로 절차적 지식을 점진적으로 생성할 수 있습니다.
- 검색과 추론을 통합하면 더 나은 근거 계획에 도움이 될 수 있습니다. 최근의 계산 심리학 모델은 기억 회상과 의사 결정의 통합된 프로세스를 암시합니다(Zhou et al., 2023a; Zhao et al., 2022). 이는 기억 검색과 전방 시뮬레이션을 엇갈리게 하는 적응 메커니즘을 통해 에이전트가 자신의 지식을 최대한 활용할 수 있음을 시사합니다.

학습: 맥락 내 학습이나 미세 조정을 넘어서 생각하는 것. CoALA의 "학습" 정의는 이러한 방법을 포함하지만, 새로운 경험이나 지식을 저장하거나 새로운 에이전트 코드를 작성하는 것까지 확장됩니다(섹션 4.5). 중요한 미래 방향은 다음과 같습니다.

- 에이전트 코드를 수정하여 메타 학습을 하면 에이전트가 더 효과적으로 학습할 수 있습니다. 예를 들어, 더 나은 검색 절차를 학습하면 에이전트가 경험을 더 잘 활용할 수 있습니다. 최근의 확장 기반 기술(Nogueira 등, 2019; Wang 등, 2023c; Tang 등, 2023a)을 사용하면 에이전트가 특정 지식이 언제 유용할지 추론하고 이를 메타데이터로 저장하여 나중에 회상할 수 있습니다. 이러한 형태의 메타 학습을 통해 에이전트는 인간이 작성한 코드를 넘어설 수 있지만 어려움과 위험으로 인해 연구가 부족합니다.
- 새로운 학습(및 탈학습) 형태에는 특정 추론 하위 작업을 위한 소규모 모델 미세 조정(Zelikman 등, 2022; Huang 등, 2022a; Ahn 등, 2022), "탈학습"을 위해 불필요한 메모리 항목 삭제(Nguyen 등, 2022c), 여러 학습 형태 간의 상호 작용 효과 연구(Tuyls 등, 2022; Park 등, 2023; Xie 등, 2023; Khattab 등, 2022) 등이 포함될 수 있습니다.

액션 공간: 외부 도구나 액션을 넘어서 생각하기. "액션 공간"은 강화 학습에서 표준 용어이지만 언어 에이전트에서는 아껴서 사용되었습니다. CoALA는 내부(추론, 검색, 학습) 및 외부(접지) 액션을 모두 포함하는 명확하고 작업에 적합한 액션 공간을 정의하는 것을 주장하며, 이는 에이전트 디자인을 체계화하고 알리는 데 도움이 될 것입니다.

- 액션 공간의 크기. 더 유능한 에이전트(예: Voyager, Generative Agents)는 더 큰 액션 공간을 가지고 있습니다. 즉, 더 복잡한 의사 결정 문제에 직면하게 됩니다. 결과적으로 이러한 에이전트는 더 맞춤화되거나 수작업으로 만든 의사 결정 절차에 의존합니다. 액션 공간 대 의사 결정 복잡성의 균형은 에이전트를 개발하기 전에 고려해야 할 기본적인 문제이며, 주어진 작업을 해결하는 데 필요한 최소한의 액션 공간을 취하는 것이 더 바람직할 수 있습니다.
- 액션 공간의 안전성. 액션 공간의 일부 부분은 본질적으로 더 위험합니다. "학습" 액션(특히 절차적 삭제 및 수정)은 내부적 피해를 일으킬 수 있는 반면, "접지" 액션(예: bash 터미널의 "rm", 인간 대화에서의 해로운 말, 물리적 환경에서 칼을 잡는 것)은 외부적 피해를 일으킬 수 있습니다. 오늘날 안전 조치는 일반적으로 작업별 휴리스틱입니다.

(예: Python에서 "os" 작업 제거(Chen et al., 2021), 대화에서 키워드 필터링(Chowdhery et al., 2022; Driess et al., 2023), 로봇을 제어된 환경으로 제한(Ahn et al., 2022)). 그러나 에이전트가 더 풍부한 내부 메커니즘을 갖춘 더 복잡한 환경에 기반을 두게 되면 최악의 시나리오 예측 및 방지를 위해 에이전트의 행동 공간을 지정하고 제거해야 할 수 있습니다(Yao and Narasimhan, 2023).

의사 결정: 행동 생성을 넘어서 생각하기. 우리는 언어 에이전트의 가장 흥미로운 미래 방향 중 하나가 의사 결정이라고 믿습니다. 섹션 4.6에서 자세히 설명했듯이, 대부분의 작업은 여전히 단일 행동을 제안(또는 직접 생성)하는 데 국한되어 있습니다. 현재 에이전트는 보다 의도적인 제안-평가-선택 의사 결정 절차의 표면만 긁은 것입니다.

- 언어 기반 추론과 코드 기반 계획을 혼합하면 두 가지의 장점을 모두 얻을 수 있습니다. 기존 접근 방식은 자연어로 직접 계획하거나(Huang et al., 2022c; Ahn et al., 2022) LLM을 사용하여 자연어에서 구조화된 세계 모델로 변환합니다(Wong et al., 2023; Liu et al., 2023a; Zhang et al., 2023a; Li et al., 2023a; Guan et al., 2023; Silver et al., 2022; 2023). 향후 작업에서는 이를 통합할 수 있습니다. Soar가 물리적 추론을 위한 시뮬레이터를 통합하는 것처럼(Laird, 2022), 에이전트는 계획의 결과를 평가하기 위해 즉석에서 시뮬레이션 코드를 작성하고 실행할 수 있습니다. 자세한 내용은 섹션 7을 참조하십시오.

- 심의 추론을 실제 환경으로 확장합니다. 초기 작업에서는 24 게임이나 블록 쌓기와 같은 장난감 과제를 사용하여 고전적인 계획 및 트리 검색(Yao et al., 2023; Hao et al., 2023; Liu et al., 2023a; Dagan et al., 2023)을 구현했습니다. 이러한 계획을 접지(Qin et al., 2023) 및 장기 기억이 있는 더 복잡한 과제로 확장하는 것은 흥미로운 방향입니다.

- 효율성을 개선하기 위한 메타 추론. LLM 호출은 느리고 계산 집약적입니다. 의사 결정에 LLM을 사용하려면 계산 비용과 결과적으로 개선된 계획의 유용성 간의 균형이 필요합니다. 대부분의 LLM 추론 방법은 추론 깊이를 지정하여 검색 예산을 고정하지만(Yao et al., 2023), 인간은 계산을 적응적으로 할당하는 것으로 보입니다(Russek et al., 2022; Lieder and Griffiths, 2020; Callaway et al., 2022; Gershman et al., 2015). 향후 작업에서는 계획의 유용성을 추정하는 메커니즘을 개발하고(Laidlaw et al., 2023) 이에 따라 의사 결정 절차를 수정해야 합니다. 이는 상각(이전 작업의 결과에 따라 LLM을 미세 조정, 예: Nguyen, 2023; Hamrick et al., 2019), 여러 의사 결정 하위 절차 간 라우팅(예: ReAct(Yao et al., 2022b)가 CoT(Wei et al., 2022b)로 후퇴하는 것을 조사함) 또는 의사 결정 절차를 업데이트하는 것입니다.

- 교정 및 정렬. 더 복잡한 의사 결정은 현재 과신 및 잘못된 교정(Jiang 등, 2021; Braverman 등, 2020; Chen 등, 2022), 인간적 가치 또는 편견과의 불일치(Liang 등, 2021; Feng 등, 2023), 자기 평가에서의 환각(Shinn 등, 2023), 불확실성에 직면한 인간 참여 메커니즘의 부족(Nguyen 등, 2022a; Ren 등, 2023)과 같은 문제로 인해 병목 현상에 처해 있습니다. 이러한 문제를 해결하면 에이전트 백본으로서 LLM의 유용성이 크게 향상될 것입니다.

7 토론

위에 제시된 실질적인 통찰력 외에도 CoALA는 여러 가지 개방형 개념적 질문을 제기합니다. 우리는 미래 연구와 토론을 위한 가장 흥미로운 중요한 방향을 간략하게 강조합니다.

LLM 대 VLM: 추론은 언어 전용이어야 할까, 아니면 다중 모드여야 할까? 대부분의 언어 에이전트는 의사 결정을 위해 언어 전용 모델을 사용하며(Yao et al., 2022b; Wang et al., 2023a; Yao et al., 2023), 필요할 때 환경 관찰을 텍스트로 변환하기 위해 별도의 캡션 모델을 사용합니다(Ahn et al., 2022; Zeng et al., 2022). 그러나 최신 세대의 언어 모델은 다중 모드이며, 이미지와 텍스트를 섞어서 입력할 수 있습니다(OpenAI, 2023a; Alayrac et al., 2022; Team et al., 2023; Li et al., 2023b). 이러한 다중 모달 모델을 기반으로 구축된 언어 에이전트는 기본적으로 이미지와 텍스트 입력에 대해 추론할 수 있습니다(Bavishi et al., 2023; Elsen et al., 2023; Liu et al., 2023b; Hong et al., 2023b; Driess et al., 2023). 이를 통해 지각적 정보를 섭취할 수 있습니다.

데이터를 직접 생성하고 작업을 직접 생성합니다. 이는 손실이 있는 이미지-텍스트 변환을 우회하지만 추론 및 계획 프로세스를 모델의 입력 모드와 긴밀하게 결합합니다.

높은 수준에서 두 가지 접근 방식은 비언어적 모달리티를 핵심 추론 모델의 언어 도메인으로 변환하기 위한 서로 다른 토큰화 체계로 볼 수 있습니다. 모듈형 접근 방식은 별도의 이미지-텍스트 모델을 사용하여 지각 데이터를 언어로 변환하는 반면(Ahn et al., 2022; Zeng et al., 2022), 통합 접근 방식은 이미지를 언어 모델의 표현 공간에 직접 투사합니다(Bavishi et al., 2023; Elsen et al., 2023; Liu et al., 2023b). 통합된 다중 모달 추론은 더 인간과 유사한 행동을 허용할 수 있습니다. VLM 기반 에이전트는 웹 페이지를 "볼" 수 있는 반면 LLM 기반 에이전트는 원시 HTML을 받을 가능성이 더 높습니다. 그러나 에이전트의 지각 및 추론 시스템을 결합하면 에이전트가 도메인에 더 특화되고 업데이트하기 어려워집니다. 어느 경우든 CoALA에서 설명하는 기본적인 아키텍처 원칙(내부 메모리, 구조화된 행동 공간, 일반화된 의사 결정)을 사용하여 에이전트 설계를 안내할 수 있습니다.

내부 대 외부: 에이전트와 환경의 경계는 무엇인가? 인간이나 로봇은 구체화된 환경과 명확히 구별되는 반면, 디지털 언어 에이전트는 경계가 덜 명확하다. 예를 들어, 위키피디아 데이터베이스는 내부 의미 메모리인가 외부 디지털 환경인가(Yao et al., 2022b)? 에이전트가 답변을 제출하기 전에 반복적으로 코드를 실행하고 개선하는 경우(Shinn et al., 2023; Yang et al., 2023), 코드 실행은 내부인가 외부인가? 방법이 제안과 평가 프롬프트로 구성된 경우(Yao et al., 2023), 단일 에이전트로 간주해야 할까, 아니면 두 개의 협력하는 더 간단한 에이전트(제안자와 평가자)로 간주해야 할까?

우리는 경계 문제가 *controllability*과 *coupling*의 관점에서 답할 수 있다고 제안합니다. 예를 들어, 위키피디아는 *controllable*이 아닙니다. 다른 사용자가 예상치 못하게 수정할 수 있는 외부 환경입니다. 그러나 에이전트만 *is*에 쓸 수 있는 오프라인 버전은 제어 가능하므로 내부 메모리로 간주할 수 있습니다. 마찬가지로 내부 가상 환경에서의 코드 실행은 내부 추론 작업으로 간주해야 하는 반면, 외부 머신(보안 취약성이 있을 수 있음)에서의 코드 실행은 외부 접지 작업으로 간주해야 합니다. 마지막으로 제안 및 평가 프롬프트와 같은 에이전트의 측면이 서로를 위해 설계되고 서로에 종속되는 경우 *tightly coupled*이며 개별 에이전트의 구성 요소로 가장 잘 개념화됩니다. 반면, 단계가 독립적으로 유용한 경우 다중 에이전트 관점이 더 적절할 수 있습니다. 이러한 딜레마는 주로 개념적이지만 이러한 이해는 에이전트 설계를 지원하고 분야가 공유 용어에 맞춰 정렬하는 데 도움이 될 수 있습니다. 실무자는 일관되고 자신의 작업에 유용한 한 선호하는 프레이밍을 선택할 수도 있습니다.

물리적 vs. 디지털: 어떤 차이가 주의를 낳을까? 동물은 물리적 세계에서 단 한 번만 살지만, 디지털 환경(예: 인터넷)은 종종 순차적(재설정을 통해)이고 병렬적인 시도를 허용합니다. 즉, 디지털 에이전트는 더 대담하게 탐색(예: 백만 개의 웹 페이지 열기)하고 병렬적인 작업 해결을 위해 자체 복제(예: 백만 개의 웹 에이전트가 다른 웹 경로 시도)할 수 있으며, 그 결과 인간의 인지에서 영감을 받은 현재의 절차와 다른 의사 결정 절차가 생길 수 있습니다(Griffiths, 2020).

학습 대 행동: 에이전트는 어떻게 지속적이고 자율적으로 학습해야 할까? CoALA 프레임워크에서 학습은 그라운딩과 마찬가지로 의사 결정 주기의 결과 행동입니다. 에이전트는 의도적으로 정보를 장기 메모리에 저장하기로 선택합니다. 이는 단순히 학습 일정을 고정하고 외부 행동에 대해서만 의사 결정을 사용하는 대부분의 에이전트와 대조적입니다. 그러나 생물학적 에이전트는 이러한 사치가 없습니다. 그들은 평생 동안 학습과 외부 행동의 균형을 맞춰야 하며 언제 무엇을 배울지 선택해야 합니다(Mattar 및 Daw, 2018). 보다 유연한 언어 에이전트(Wang et al., 2023a; Park et al., 2023)는 유사한 설계를 따르고 학습을 외부 행동과 동등하게 취급합니다. 학습은 정기적인 의사 결정 중에 가능한 행동으로 제안될 수 있으며, 에이전트는 적절한 시간까지 학습을 "연기"할 수 있습니다.

GPT-4 대 GPT-N: 더 강력한 LLM으로 에이전트 설계가 어떻게 바뀔까? 에이전트 설계는 규모에 따라 새로운 LLM 역량이 등장함에 따라 움직이는 표적입니다(Wei et al., 2022a). 예를 들어, GPT-2(Radford et al., 2019)와 같은 초기 언어 모델은 LLM 에이전트를 지원하지 않았습니다. 실제로 당시 작업에는 GPT-2를 강화 학습과 결합하여 액션을 생성해야 했습니다(Yao et al., 2020). GPT-3(Brown et al., 2020)은 NLP 작업을 위한 유연한 few-shot 및 zero-shot 추론을 잠금 해제했습니다. GPT-4(OpenAI, 2023a)만이 더 신뢰할 수 있는 자체 평가(Saunders et al., 2022; Shinn et al., 2023; Yao et al., 2023)와 자체 정제(Madaan et al., 2023; Chen et al., 2023b)를 제공하기 시작했습니다. 향후 LLM은 코드화된 규칙과 추가 학습 모델의 필요성을 더욱 줄일 수 있을까요? 이로 인해 CoALA 프레임워크를 변경해야 할까요? 사고 실험으로 GPT-N이 기억, 접지, 학습을 "시뮬레이션"할 수 있다고 상상해 보세요.

맥락에서의 의사결정: 가능한 모든 행동을 나열하고, 각각을 시뮬레이션하고 평가하고, 매우 긴 맥락에서 전체 장기 메모리를 명시적으로 유지합니다. 또는 더 대담하게 말해서, 아마도 GPT-N+1은 맥락에서 중간 추론 없이 뉴런에서 암묵적으로 이를 시뮬레이션하여 다음 행동을 생성하는 데 성공할 것입니다. 이러한 극단적인 사례는 당장은 불가능해 보이지만 점진적인 개선으로 인해 다양한 CoALA 구성 요소의 중요성이 변경될 수 있습니다. 예를 들어, 더 긴 맥락 창은 장기 메모리의 중요성을 줄일 수 있는 반면, 내부 평가 및 시뮬레이션에 대한 더 강력한 추론은 더 긴 지평선 계획을 허용할 수 있습니다. 일반적으로 LLM은 생물학적 한계의 영향을 받지 않으며(Griffiths, 2020), 그 출현 속성은 예측하기 어렵습니다. 그럼에도 불구하고 CoALA(그리고 보다 일반적으로 인지 과학)는 언어 에이전트가 성공하거나 실패하는 작업을 구성하는 데 여전히 도움이 될 수 있으며, 주어진 작업에서 주어진 LLM을 보완하는 코드 기반 절차를 제안할 수 있습니다. GPT가 CoALA의 모든 메커니즘을 뉴런에 구현하는 가장 극단적인 경우에도 CoALA를 개념적 가이드로 활용하여 이러한 암묵적 회로를 발견하고 해석하는 것이 도움이 될 수 있습니다. 물론, 섹션 6에서 논의했듯이 에이전트 사용 사례도 LLM 기능을 발견하고 정의하고 형성하는 데 도움이 됩니다. 칩과 컴퓨터 아키텍처가 공동 진화한 방식과 유사하게 언어 모델과 에이전트 설계도 상호 경로를 개발해야 합니다.

8 결론

우리는 언어 에이전트를 설명하고 구축하기 위한 개념적 프레임워크인 CoALA(Cognitive Architectures for Language Agents)를 제안했습니다. 우리의 프레임워크는 상징적 인공 지능과 인지 과학의 풍부한 역사에서 영감을 얻어 수십 년 된 통찰력을 대규모 언어 모델에 대한 최첨단 연구에 연결합니다. 우리는 이 접근 방식이 보다 일반적이고 인간과 더 유사한 인공 지능을 개발하는 길을 제공한다고 믿습니다.

감사의 말

교정과 귀중한 피드백을 해주신 Harrison Chase, Baian Chen, Khanh Nguyen, Ofir Press, Noah Shinn, Jens Tuyls, 그리고 도움이 되는 토론을 해주신 Princeton NLP Group과 Princeton Computational Cognitive Science Lab의 구성원들에게 감사드립니다. 마지막으로, 통찰력 있는 의견과 제안을 해주신 익명의 심사위원들에게 감사드립니다. SY와 KN은 Oracle Collaborative Research 상과 Grant No. 2239363에 따른 National Science Foundation의 지원에 감사드립니다. 이 자료에 표현된 모든 의견, 결과, 결론 또는 권장 사항은 저자의 것이며 반드시 National Science Foundation의 견해를 반영하는 것은 아닙니다. SY는 또한 Princeton의 Harold W. Dodds Fellowship의 지원을 받고 있습니다. TS는 National Defense Science and Engineering(NDSEG) Graduate Fellowship Program의 지원을 받고 있습니다.

참고문헌

S. Adams, I. Arel, J. Bach, R. Coop, R. Furlan, B. Goertzel, J. S. Hall, A. Samsonovich, M. Scheutz, M. Schlesinger 등. 인간 수준의 인공 일반 지능의 풍경 매핑. *AI magazine*, 33(1):25 – 42, 2012. M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman 등. 내가 말하는 대로 하지 말고 내가 할 수 있는 대로 하라: 로봇의 가능성에 기반을 둔 언어. *arXiv preprint arXiv:2204.01691*, 2022. J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: few-shot learning을 위한 시각적 언어 모델. *Advances in Neural Information Processing Systems*, 35:23716 – 23736, 2022. J. R. Anderson 및 C. Lebiere. 인지 이론에 대한 Newell 테스트. *Behavioral and Brain Sciences*, 26(5):587 – 601, 2003. J. Andreas. 에이전트 모델로서의 언어 모델.
Findings of the Association for Computational Linguistics: EMNLP 2022, 5769 – 5779페이지, 2022. R. C. Atkinson 및 R. M. Shiffrin. 인간 기억: 제안된 시스템 및 제어 프로세스. *Psychology of Learning and Motivation*, 2권, 89 – 195페이지. 엘스비어, 1968.

A. D. Baddeley와 G. Hitch. 작업 기억. *Psychology of Learning and Motivation*, 8권, 47-89쪽. Elsevier, 1974.

Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon 등 헌법적 AI: AI 피드백이 무해합니다. *arXiv preprint arXiv:2212.08073*, 2022.

R. Bavishi, E. Elsen, C. Hawthorne, M. Nye, A. Odena, A. Somani 및 S. Ta ş ı rlar. 2023년 다중 모드 모델을 소개합니다. URL `https://www.adept.ai/blog/fuyu-8b`.

Y. Bisk, D. Marcu 및 W. Wong. 근거 언어 습득을 통한 인간 컴퓨터 의사소통을 위한 데이터 세트를 지향합니다. *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016년.

E. Biyik 및 M. Palan. 쉬운 질문하기: 능동적 보상 학습에 대한 사용자 친화적 접근 방식.
Proceedings of the 3rd Conference on Robot Learning, 2019.

C. Blundell, B. Uria, A. Pritzel, Y. Li, A. Ruderman, J. Z. Leibo, J. Rae, D. Wierstra 및 D. Hassabis. 모델이 없는 에피소드 제어. *arXiv preprint arXiv:1606.04460*, 2016.

S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. B. Van Den Driessche, J.-B. Lespiau, B. Damoc, A. Clark 등. 수조 개의 토큰을 검색하여 언어 모델을 개선합니다.
International Conference on Machine Learning, 페이지 2206 – 2240, 2022.

S. Branavan, D. Silver, R. Barzilay. Monte-Carlo 프레임워크에서 매뉴얼을 읽어서 이기는 법 배우기.
Journal of Artificial Intelligence Research, 43:661 – 704, 2012.

M. Braverman, X. Chen, S. Kakade, K. Narasimhan, C. Zhang, and Y. Zhang. 언어 모델의 교정, 엔트로피 비율 및 메모리. *International Conference on Machine Learning*, 1089-1099쪽, 2020.

G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba. Openai 체육관, 2016.

A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. RT-1: 규모에 따른 실제 세계 제어를 위한 로봇 트랜스포머. *arXiv preprint arXiv:2212.06817*, 2022.

A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. RT-2: 시각-언어-행동 모델이 웹 지식을 로봇 제어로 전환합니다. *arXiv preprint arXiv:2307.15818*, 2023.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. 언어 모델은 few-shot 학습자입니다. *Advances in Neural Information Processing Systems*, 33:1877 – 1901, 2020.

C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, 및 S. Colton. 몬테카를로 트리 탐색 방법 조사. *IEEE Transactions on
Computational Intelligence and AI in games*, 4(1):1 – 43, 2012.

F. Callaway, B. van Opheusden, S. Gul, P. Das, P. M. Krueger, T. L. Griffiths, 및 F. Lieder. 인간 계획에서 인지 자원의 합리적 사용. *Nature Human Behaviour*, 6(8):1112 – 1125, 2022.

C.-M. Chan, W. Chen, Y. Su, J. Yu, W. Xue, S. Zhang, J. Fu, and Z. Liu. Chateval: 다중 에이전트 토론을 통한 더 나은 LLM 기반 평가자를 향해. *arXiv preprint arXiv:2308.07201*, 2023. B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and D. Kappler. 실제 세계 계획을 위한 개방형 어휘 쿼리 가능 장면 표현. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 11509 – 11522쪽, 2023a.

D. Chen과 R. Mooney. 관찰을 통해 자연어 항해 지침을 해석하는 법 배우기.
Proceedings of the AAAI Conference on Artificial Intelligence, 25권, 859~865페이지, 2011년.

D. Chen, A. Fisch, J. Weston, A. Bordes. 오픈 도메인 질문에 답하기 위해 위키피디아 읽기. *arXiv preprint arXiv:1704.00051*, 2017.

M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman 등. 코드로 학습된 대규모 언어 모델 평가. *arXiv preprint arXiv:2107.03374*, 2021.

X. Chen, M. Lin, N. Schärli, and D. Zhou. 자체 디버깅을 위한 대규모 언어 모델 교육. *arXiv preprint arXiv:2304.05128*, 2023b. Y. Chen, L. Yuan, G. Cui, Z. Liu, and H. Ji. 사전 학습된 언어 모델의 교정에 대한 자세한 고찰. *arXiv preprint arXiv:2211.00151*, 2022.

N. Chomsky. 언어 설명을 위한 세 가지 모델. *IRE Transactions on information theory*, 2(3): 113 – 124, 1956.

A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: 경로로 언어 모델링 확장. *arXiv preprint arXiv:2204.02311*, 2022.

P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, D. Amodei. 인간 선호도에서 심층 강화 학습.
Advances in neural information processing systems, 30, 2017.

A. 교회. 논리의 기초를 위한 일련의 가정. *Annals of mathematics*, 346-366쪽, 1932년.

M.-A. Côté, A. Kádár, X. Yuan, B. Kybartas, T. Barnes, E. Fine, J. Moore, M. Hausknecht, L. El Asri, M. Adada, et al. Textworld: 텍스트 기반 게임을 위한 학습 환경. *Computer Games: 7th Workshop, CGW 2018*, 41-75페이지. Springer, 2019.

A. Creswell, M. Shanahan, I. Higgins. 선택 추론: 해석 가능한 논리적 추론을 위한 대규모 언어 모델 활용.
The Eleventh International Conference on Learning Representations, 2023.

G. Dagan, F. Keller, A. Lascarides. LLM을 통한 동적 계획. *arXiv preprint arXiv:2308.06391*, 2023.

I. Dasgupta, C. Kaeser-Chen, K. Marino, A. Ahuja, S. Babayan, F. Hill, R. Fergus. 체화된 추론을 위한 언어 모델과의 협업. *Second Workshop on Language and Reinforcement Learning*, 2022.

X. Deng, Y. Gu, B. Zheng, S. Chen, S. Stevens, B. Wang, H. Sun, and Y. Su. Mind2Web: 웹을 위한 일반 에이전트를 향하여. *arXiv preprint arXiv:2306.06070*, 2023.

N. Derbinsky, J. Li, and J. Laird. A multi-domain evaluation of scaling in a general episodic memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 193 – 199, 2012. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: 언어 이해를 위한 딥 양방향 트랜스포머의 사전 학습. In *NAACL-HLT (1)*, 2019. D. Dohan, W. Xu, A. Lewkowycz, J. Austin, D. Bieber, R. G. Lopes, Y. Wu, H. Michalewski, R. A. Saurous, J. Sohl-Dickstein, et al. 언어 모델 캐스케이드. *arXiv preprint arXiv:2207.10342*, 2022. Y. Dong, X. Jiang, Z. Jin, and G. Li. chatgpt를 통한 자체 협업 코드 생성. *arXiv preprint arXiv:2304.07590*, 2023. D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, 외. Palm-e: 구현된 다중 모드 언어 모델입니다. *arXiv preprint arXiv:2303.03378*, 2023.

Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, I. Mordatch. 다중 에이전트 토론을 통해 언어 모델의 사실성과 추론 개선. *arXiv preprint arXiv:2305.14325*, 2023.

A. Ecoffet, J. Huizinga, J. Lehman, K. O. Stanley, J. Clune. Go-explore: 하드 탐색 문제에 대한 새로운 접근 방식. *arXiv preprint arXiv:1901.10995*, 2019.

K. Ellis, C. Wong, M. Nye, M. Sablé-Meyer, L. Morales, L. Hewitt, L. Cary, A. Solar-Lezama, J. B. Tenenbaum. Dre amcoder: 깨어 있는-잠자는 라이브러리 학습을 통한 귀납적 프로그램 합성 부트스트래핑. *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, 835-850페이지, 2021. E. Elsen, A. Odena, M. Nye, S. Taş ı rlar, T. Dao, C. Hawthorne, D. Mop arthi, A. Somani. Persimmon-8B 출시, 2023. URL `https://www.adept.ai/blog/persimmon-8b`. S. Feng, C. Y. Park, Y. Liu, Y. Tsvetkov. 사전 학습 데이터에서 언어 모델, 다운스트림 작업으로: 불공정한 NLP 모델로 이어지는 정치적 편견의 흔적 추적. *arXiv preprint arXiv:2305.08283*, 2023.

D. Ganguli, A. Askell, N. Schiefer, T. Liao, K. Luko š iūtė, A. Chen, A. Goldie, A. Mirhoseini, C. Olsson, D. Hernan dez 등 . 대규모 언어 모델의 도덕적 자기 교정 능력. *arXiv preprint arXiv:2302.07459*, 2023.

C. Gao, X. Lan, Z. Lu, J. Mao, J. Piao, H. Wang, D. Jin, and Y. Li. S3: 대규모 언어 모델이 강화된 에이전트를 갖춘 소셜 네트워크 시뮬레이션 시스템. *arXiv preprint arXiv:2307.14984*, 2023.

T. Gao, A. Fisch, D. Chen. 사전 훈련된 언어 모델을 더 나은 few-shot 학습자로 만들기. *arXiv preprint arXiv:2012.15723*, 2020.

S. J. Gershman, E. J. Horvitz, J. B. Tenenbaum. 계산적 합리성: 뇌, 정신, 기계의 지능을 위한 수렴 패러다임. *Science*, 349(6245):273 – 278, 2015.

T. L. Griffiths. 인간의 한계를 통해 인간 지능 이해. *Trends in Cognitive Sciences*, 24(11):873 – 883, 2020.

J. Gu, Y. Wang, K. Cho, 및 V. O. Li. 검색 엔진 유도 신경 기계 번역. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32권, 2018년.

L. Guan, K. Valmeekam, S. Sreedharan, S. Kambhampati. 사전 훈련된 대규모 언어 모델을 활용하여 모델 기반 작업 계획을 위한 세계 모델을 구성하고 활용합니다. *arXiv preprint arXiv:2305.14909*, 2023.

지침. 지침, 2023. URL `https://github.com/guidance-ai/guidance`.

I. Gur, H. Furuta, A. Huang, M. Safdari, Y. Matsuo, D. Eck, and A. Faust. 계획, 긴 맥락 이해 및 프로그램 합성을 갖춘 실제 웹 에이전트. *arXiv preprint arXiv:2307.12856*, 2023.

K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang. 검색 증강 언어 모델 사전 학습. *International conference on machine learning*, 3929 – 3938쪽, 2020.

J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, T. Pfaff, T. Weber, L. Buesing, P. W. Battaglia. q-learning과 검색을 상각 가치 추정치와 결합. *International Conference on Learning Representations*, 2019.

A. W. Hanjie, V. Zhong, K. Narasimhan. 강화 학습에서 일반화를 위한 엔티티와 역학에 대한 언어 접지. *International Conference on Machine Learning (ICML)*, 2021.

S. Hao, Y. Gu, H. Ma, J. J. Hong, Z. Wang, D. Z. Wang, Z. Hu. 언어 모델을 통한 추론은 세계 모델을 통한 계획입니다. *arXiv preprint arXiv:2305.14992*, 2023.

M. Hasan, C. Ozel, S. Potter, E. Hoque. Sapien: 대규모 언어 모델로 구동되는 정서적 가상 에이전트. *arXiv preprint arXiv:2308.03022*, 2023.

P. Haslum, N. Lipovetzky, D. Magazeni, C. Muise, R. Brachman, F. Rossi 및 P. Stone. *An introduction to the planning domain definition language*, 13권. 스프링거, 2019.

M. Hausknecht, P. Ammanabrolu, M.-A. Côté, X. Yuan. 대화형 소설 게임: 거대한 모험.
Proceedings of the AAAI Conference on Artificial Intelligence, 34권, 7903~7910페이지, 2020년.

S. Hong, X. Zheng, J. Chen, Y. Cheng, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou, C. Ran, et al. Metagpt: 다중 에이전트 협업 프레임워크를 위한 메타 프로그래밍. *arXiv preprint arXiv:2308.00352*, 2023a. W. Hong, W. Wang, Q. Lv, J. Xu, W. Yu, J. Ji, Y. Wang, Z. Wang, Y. Dong, M. Ding, et al. Cogagent: GUI 에이전트를 위한 시각적 언어 모델. *arXiv preprint arXiv:2312.08914*, 2023b. J. Huang, S. S. Gu, L. Hou, Y. Wu, X. Wang, H. Yu, and J. Han. 대규모 언어 모델은 자체 개선이 가능합니다. *arXiv preprint arXiv:2210.11610*, 2022a. S. Huang, Z. Jiang, H. Dong, Y. Qiao, P. Gao, and H. Li. Instruct2Act: 대규모 언어 모델을 사용하여 다중 모달 지침을 로봇 동작에 매핑합니다. *arXiv preprint arXiv:2305.11176*, 2023.

W. Huang, P. Abbeel, D. Pathak, I. Mordatch. 제로샷 플래너로서의 언어 모델: 구체화된 에이전트를 위한 실행 가능한 지식 추출. *International Conference on Machine Learning*, 9118-9147페이지, 2022b.

W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, et al. 내면 독백: 언어 모델을 통한 계획을 통한 구체화된 추론. *arXiv preprint arXiv:2207.05608*, 2022c.

A. Hussein, M. M. Gaber, E. Elyan 및 C. Jayne. 모방 학습: 학습 방법 조사. *ACM Computing Surveys (CSUR)*, 50(2):1 – 35, 2017.

G. Irving, P. Christiano, D. Amodei. 토론을 통한 AI 안전. *arXiv preprint arXiv:1805.00899*, 2018.

G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave. 대조 학습을 통한 비지도 고밀도 정보 검색. *arXiv preprint arXiv:2112.09118*, 2021.

Z. Jiang, J. Araki, H. Ding, G. Neubig. 언어 모델이 언제 아는지 어떻게 알 수 있을까? 질문 답변을 위한 언어 모델 교정에 관하여. *Transactions of the Association for Computational Linguistics*, 9:962 – 977, 2021. Z. Jin, S. Levine, F. G. Adauto, O. Kamal, M. Sap, M. Sachan, R. Mihalcea, J. B. Tenenbaum, B. Schölkopf. 예외를 만드는 경우: 인간의 도덕적 판단에 대한 설명으로서 언어 모델 탐구. A. H. Oh, A. Agarwal, D. Belgrave, K. Cho 편집자, *Advances in Neural Information Processing Systems*, 2022. S. Jinxin, Z. Jiabao, W. Yilei, W. Xingjiao, L. Jiawen, H. Liang. Cgmi: 구성 가능한 일반 다중 에이전트 상호 작용 프레임워크.
arXiv preprint arXiv:2308.12503, 2023. R. M. Jones, J. E. Laird, P. E. Nielsen, K. J. Coulter, P. Kenny, F. V. Koss. 전투 비행 시뮬레이션을 위한 자동화된 지능형 조종사. *AI magazine*, 20(1):27 – 27, 1999. D. Jurafsky.
Speech & language processing. Pearson Education India, 2000. O. Khattab, K. Santhanam, X. L. Li, D. Hall, P. Liang, C. Potts, M. Zaharia. 시연-검색-예측: 지식 집약적 NLP를 위한 검색 및 언어 모델 구성.
arXiv preprint arXiv:2212.14024, 2022. URL `https://github.com/stanfordnlp/dspy`. G. Kim, P. Baldi, S. McAleer. 언어 모델은 컴퓨터 작업을 해결할 수 있습니다. *arXiv preprint arXiv:2303.17491*, 2023. J. R. Kirk 및 J. E. Laird. 간단한 게임을 위한 대화형 과제 학습. *Advances in Cognitive Systems*, 3(13-30):5, 2014.

J. R. Kirk, W. Robert, P. Lindes, and J. E. Laird. 에이전트 분석을 통한 로봇 작업 학습을 위한 LLM에서 지식 추출 개선. *arXiv preprint arXiv:2306.06770*, 2023. K. R. Koedinger, J. R. Anderson, W. H. Hadley, M. A. Mark, et al. 지능형 튜터링이 대도시 학교에 간다. *International Journal of Artificial Intelligence in Education*, 8(1):30 – 43, 1997. T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. 대규모 언어 모델은 제로샷 추론기이다. *Advances in Neural Information Processing Systems*, 35:22199 – 22213, 2022. I. Kotseruba and J. K. Tsotsos. 40 년간의 인지 아키텍처: 핵심 인지 능력과 실용적인 응용 프로그램. *Artificial Intelligence Review*, 53(1):17 – 94, 2020. C. Laidlaw, S. Russell, 및 A. Dragan. 효과적인 지평을 통해 rl 이론과 실천을 연결하기. *arXiv preprint arXiv:2304.09853*, 2023. J. E. Laird. *The Soar cognitive architecture*. MIT 출판부, 2019. J. E. Laird. Soar 소개. *arXiv preprint arXiv:2205.03854*, 2022.

J. E. Laird, P. S. Rosenbloom, A. Newell. Soar에서의 Chunking: 일반 학습 메커니즘의 해부학. *Machine Learning*, 1:11 – 46, 1986.

J. E. Laird, A. Newell 및 P. S. Rosenbloom. Soar: 일반 지능을 위한 아키텍처. *Artificial Intelligence*, 33(1):1 – 64, 1987.

J. E. Laird, K. R. Kinkade, S. Mohan, J. Z. Xu. Soar 인지 아키텍처를 사용한 인지 로봇. *CogRob @ AAAI*, 2012.

B. M. Lake, T. D. Ullman, J. B. Tenenbaum, S. J. Gershman. 사람처럼 배우고 생각하는 기계 만들기, 2016.

랭체인. 랭체인, 2022. URL `http://www.langchain.com`.

H. Le, Y. Wang, A. D. Gotmare, S. Savarese, S. C. H. Hoi. Coderl: 사전 학습된 모델과 심층 강화 학습을 통한 코드 생성 마스터링. *Advances in Neural Information Processing Systems*, 35:21314 – 21328, 2022.

Y. LeCun. 자율 기계 지능을 향한 길 버전 0.9.2, 2022-06-27. *Open Review*, 62, 2022.

P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. 지식 집약적 NLP 작업을 위한 검색 증강 생성. *Advances in Neural Information Processing Systems*, 33:9459 – 9474, 2020. B. Z. Li, W. Chen, P. Sharma, and J. Andreas. Lampp: 지각과 행동에 대한 확률적 사전 확률로서의 언어 모델. *arXiv preprint arXiv:2302.02801*, 2023a. C. Li, Z. Gan, Z. Yang, J. Yang, L. Li, L. Wang, and J. Gao. 다중 모달 기반 모델: 전문가에서 일반 보조자로. *arXiv preprint arXiv:2309.10020*, 2023b. H. Li, Y. Su, D. Cai, Y. Wang 및 L. Liu. 검색증강 텍스트 생성에 관한 조사. *arXiv preprint arXiv:2202.01110*, 2022a. R. Li, L. B. Allal, Y. Zi, N. Muennighoff, D. Kocetkov, C. Mou, M. Marone, C. Akiki, J. Li, J. Chim, Q. Liu, E. Zheltonozhskii, T. Y. Zhuo, T. Wang, O. Dehaene, M. Davaadorj, J. Lamy-Poirier, J. Monteiro, O. Shliazhko, N. Gontier, N. 미드, A. Zebaze, M.-H. Yee, L. K. Umapathi, J. Zhu, B. Lipkin, M. Oblokulov, Z. Wang, R. Murthy, J. Stillerman, S. S. Patel, D. Abulkhanov, M. Zocca, M. Dey, Z. Zhang, N. Fahmy, U. Bhattacharyya, W. Yu, S. Singh, S. Luccioni, P. Villegas, M. Kunakov, F. Zhdanov, M. Romero, T. Lee, N. Timor, J. Ding, C. Schlesinger, H. Schoelkopf, J. Ebert, T. Dao, M. Mishra, A. Gu, J. Robinson, C. J. Anderson, B. Dolan-Gavitt, D. Contractor, S. Reddy, D. Fried, D. Bahdanau, Y. Jernite, C. M. Ferrandis, S. M. 휴즈, T. Wolf, A. Guha, L. von Werra, 및 H. de Vries. Starcoder: 출처가 당신과 함께 하기를! *ArXiv*, abs/2305.06161, 2023c.

Y. Li, D. H. Choi, J. Chung, N. Kushman, J. Schrittwieser, R. Leblond, Tom, Eccles, J. Keeling, F. Gimeno, A. D. Lago, T. Hubert, P. Choy, C. de, M. d'Autume, I. Babuschkin, X. Chen, P.-S. Huang, J. Welbl, S. Gowal, Alexey, Cherepanov, J. Molloy, D. J. Mankowitz, E. S. Robson, P. Kohli, N. de, Freitas, K. Kavukcuoglu 및 O. Vinyals. 알파코드를 사용한 경쟁 수준 코드 생성. *Science*, 378:1092 – 1097, 2022b.

J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, A. Zeng. 정책으로서의 코드: 구체화된 제어를 위한 언어 모델 프로그램. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 9493 – 9500쪽, 2023a.

P. P. Liang, C. Wu, L.-P. Morency, R. Salakhutdinov. 언어 모델에서 사회적 편견을 이해하고 완화하기 위해. *International Conference on Machine Learning*, 6565-6576쪽, 2021.

T. Liang, Z. He, W. Jiao, X. Wang, Y. Wang, R. Wang, Y. Yang, Z. Tu, and S. Shi. 다중 에이전트 토론을 통해 대규모 언어 모델에서 발산적 사고를 장려합니다. *arXiv preprint arXiv:2305.19118*, 2023b.

F. Lieder와 T. L. Griffiths. 자원 합리적 분석: 제한된 계산 자원의 최적 사용으로서의 인간 인지 이해. *Behavioral and Brain Sciences*, 43:e1, 2020.

B. Y. Lin, Y. Fu, K. Yang, P. Ammanabrolu, F. Brahman, S. Huang, C. Bhagavatula, Y. Choi, and X. Ren. Swiftsage: 복잡한 상호작용 작업을 위한 빠르고 느린 사고를 가진 생성 에이전트. *arXiv preprint arXiv:2305.17390*, 2023.

P. Lindes와 J. E. Laird. 인지 언어학과 인지 언어 처리를 통합하기 위해. *Proceedings of the 14th International Conference on Cognitive Modeling (ICCM)*, 2016.

B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, P. Stone. LLM+P: 최적의 계획 능력을 갖춘 대규모 언어 모델 강화. *arXiv preprint arXiv:2304.11477*, 2023a.

H. Liu, C. Li, Q. Wu, 및 Y. J. Lee. 시각 지시 튜닝. *NeurIPS*, 2023b.

H. Liu, C. Sferrazza, P. Abbeel. 언어는 보상입니다: 인간 피드백을 사용한 후견지식 미세 조정. *arXiv preprint arXiv:2302.02676*, 2023c.

J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, W. Chen. GPT-3에 대한 좋은 맥락적 예시는 무엇인가? *arXiv preprint arXiv:2101.06804*, 2021.

P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig. 사전 학습, 프롬프트 및 예측: 자연어 처리에서 프롬프트 방법에 대한 체계적 조사. *ACM Computing Surveys*, 55(9), 2023d. ISSN 0360-0300.

R. Liu, J. Wei, S. S. Gu, T.-Y. Wu, S. Vosoughi, C. Cui, D. Zhou, A. M. Dai. 마음의 눈: 시뮬레이션을 통한 근거 언어 모델 추론. *The Eleventh International Conference on Learning Representations*, 2023e.

R. Liu, R. Yang, C. Jia, G. Zhang, D. Zhou, A. M. Dai, D. Yang, and S. Vosoughi. 시뮬레이션된 인간 사회에서 사회적으로 정렬된 언어 모델 훈련. *arXiv preprint arXiv:2305.16960*, 2023f.

LlamaIndex. LlamaIndex, 2023. URL `http://www.llamaindex.ai`.

L. E. Lwakatare, A. Raj, I. Crnkovic, J. Bosch, and H. H. Olsson. 실제 산업 환경에서의 대규모 머신 러닝 시스템: 과제와 해결책 검토. *Information and software technology*, 127:106368, 2020.

Z. Ma, Y. Mei, Z. Su. 정신적 웰빙 지원을 위한 대규모 언어 모델 기반 대화 에이전트 사용의 이점과 과제 이해. *arXiv preprint arXiv:2307.15810*, 2023.

S. Macenski, T. Foote, B. Gerkey, C. Lalancette, 및 W. Woodall. 로봇 운영 체제 2: 야생에서의 설계, 아키텍처 및 사용. *Science Robotics*, 7(66):eabm6074, 2022.

A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang 등 S elf-refine: 자체 피드백을 통해 반복적으로 개선합니다. *arXiv preprint arXiv:2303.17651*, 2023.

A. A. Markov. 알고리즘 이론. *Trudy Matematicheskogo Instituta Imeni VA Steklova*, 42:3 – 375, 1954.

M. G. Mattar 및 N. D. Daw. 우선순위가 지정된 메모리 액세스는 계획 및 해마 재생을 설명합니다. *Nature Neuroscience*, 21(11):1609 – 1617, 2018.

J. L. McClelland, F. Hill, M. Rudolph, J. Baldridge, H. Schütze. 기계 언어 모델을 인간 수준의 언어 이해로 확장. *arXiv preprint arXiv:1912.05877*, 2019.

J. Meier, R. Rao, R. Verkuil, J. Liu, T. Sercu, A. Rives. 언어 모델은 단백질 기능에 대한 돌연변이의 영향에 대 한 제로샷 예측을 가능하게 합니다. *bioRxiv*, 2021.

G. Mialon, R. Dessì, M. Lomeli, C. Nalmpantis, R. Pasunuru, R. Raileanu, B. Rozière, T. Schick, J. Dwivedi- Yu, A. Celikyilmaz 등 증강 언어 모델: 설문조사. *arXiv preprint arXiv:2302.07842*, 2023.

S. Mohan 및 J. Laird. 상황적 상호작용적 지시에서 학습 목표 지향적 계층적 과제.
Proceedings of the AAAI Conference on Artificial Intelligence, 28권, 2014년.

S. Mohan, A. H. Mininger, J. R. Kirk, J. E. Laird. 상황에 맞는 상호작용 교육을 통해 단어의 근거 표현 습득. *Advances in Cognitive Systems*, 2:113 – 130, 2012.

R. 나카노, J. 힐튼, S. 발라지, J. 우, L. 오우양, C. 킴, C. 헤세, S. 자인, V. 코사라주, W. 손더스 외. WebGPT: 인 간 피드백을 통한 브라우저 지원 질의응답. *arXiv preprint arXiv:2112.09332*, 2021.

K. Narasimhan, R. Barzilay, and T. Jaakkola. 언어 기반 강화 학습에서의 심층 전이.
Journal of Artificial Intelligence Research (JAIR), 2018.

A. Narayan-Chen, P. Jayannavar, J. Hockenmaier. Minecraft에서의 협력적 대화. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5405-5415쪽. Association for Co mputational Linguistics, 2019.

S. Nason 및 J. E. Laird. Soar-RL: 강화 학습을 Soar와 통합. *Cognitive Systems Research*, 6(1):51 – 59, 2005.

A. Newell. 문제 해결 연구: 암호 산술 과제에 대한 주제 3 DONALD+ GERALD= ROBERT. 기술 보고서, Car negie Mellon University, 1967.

A. Newell. 물리적 기호 시스템. *Cognitive science*, 4(2):135 – 183, 1980.

A. Newell. 인지의 통합 이론에 대한 요약. *Behavioral and Brain Sciences*, 15(3):425 – 437, 1992.

A. Newell 및 H. A. Simon. *Human problem solving*. Prentice-Hall, 1972.

A. Newell, P. S. Rosenbloom, J. E. Laird. 인지를 위한 상징적 아키텍처. *Foundations of cognitive science*, 93-13 1쪽, 1989.

K. Nguyen과 H. Daumé III. 도와주세요, Anna! 회고적 호기심을 북돋우는 모방 학습을 통한 자연스러운 다중 모드 지원으로 시각적 탐색. *arXiv preprint arXiv:1909.01871*, 2019.

K. Nguyen, D. Dey, C. Brockett, B. Dolan. 간접 개입을 통한 모방 학습을 통한 언어 기반 지원과 함께 시각 기 반 탐색. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12527 – 1 2537쪽, 2019.

K. Nguyen, Y. Bisk, 및 H. Daumé III. 인간에게 풍부하고 상황에 맞는 유용한 정보를 요청하는 방법을 학습하 기 위한 프레임워크. *ICML*, 2022년 7월a.

K. X. Nguyen. 언어 모델은 제한된 실용적 화자입니다. *First Workshop on Theory of Mind in Communicating Agents*, 2023.

K. X. Nguyen, D. Misra, R. Schapire, M. Dudík, and P. Shafto. 활동 설명에서 얻은 대화형 학습. *International Conference on Machine Learning*, 8096 – 8108페이지, 2021. K. X. Nguyen, Y. Bisk, and H. D. Iii. 인간에게 풍부하고 상황에 맞는 유용한 정보를 요청하는 학습 프레임워크. *International Conference on Machine Learning*, 16553 – 16568페이지, 2022b.

T. T. Nguyen, T. T. Huynh, P. L. Nguyen, A. W.-C. Liew, H. Yin, Q. V. H. Nguyen. 기계 언러닝에 대한 조사. *arXiv preprint arXiv:2209.02299*, 2022c.

A. Ni, S. Iyer, D. Radev, V. Stoyanov, W.-t. Yih, S. Wang, X. V. Lin. Lever: 실행을 통한 언어-코드 생성 검증 학습. *International Conference on Machine Learning*, 26106 – 26128페이지, 2023. N. J. Nilsson. 로봇 셰이키. *Technical Note*, 1984. R. Nogueira, W. Yang, J. Lin, K. Cho. 쿼리 예측을 통한 문서 확장, 2019.

A. M. Nuxoll과 J. E. Laird. 에피소드 기억을 통한 인지 구조 확장. *Proceedings of the AAAI Conference on Artificial Intelligence*, 1560-1564쪽, 2007.

M. Nye, A. J. Andreassen, G. Gur-Ari, H. Michalewski, J. Austin, D. Bieber, D. Dohan, A. Lewkowycz, M. Bosma, D. Luan 등. 작업 표시: 언어 모델을 사용한 중간 계산을 위한 스크래치패드. *arXiv preprint arXiv:2112.00114* , 2021.

OpenAI. Gpt-4 기술 보고서. *ArXiv*, abs/2303.08774, 2023a.

OpenAI. 함수 호출 및 기타 API 업데이트, 2023b. URL `https://openai.com/blog/function-calling-and-other-api-updates`.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray 등. 인간 피드백을 통해 지시를 따르도록 언어 모델을 훈련합니다. *Advances in Neural Information Processing Systems*, 35:27730 – 27744, 2022.

A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. Piramuthu, G. Tur, D. Hakkani-Tur. Teach: 채팅하는 작업 중심의 구체화된 에이전트. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36권, 2017-2025, 2022쪽.

N. D. Palo, A. Byravan, L. Hasenclever, M. Wulfmeier, N. Heess, and M. Riedmiller. 기초 모델을 갖춘 통합 에이전트를 향하여. *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023.

A. Parisi, Y. Zhao, and N. Fiedel. Talm: 도구 증강 언어 모델. *arXiv preprint arXiv:2205.12255*, 2022. J. S. Park, J. C. O ' Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. 생성 에이전트: 인간 행동의 대화형 시뮬라크라. *arXiv preprint arXiv:2304.03442*, 2023. P. Pataranutaporn, V. Danry, J. Leong, P. Punpongsanon, D. Novy, P. Maes, and M. Sra. 개인화된 학습과 웰빙을 지원하기 위한 AI 생성 캐릭터. *Nature Machine Intelligence*, 3(12):1013 – 1022, 2021. A. Peng, I. Sucholutsky, B. Li, T. R. Sumers, T. L. Griffiths, J. Andreas, and J. A. Shah. 언어 가이드 상태 추상화. *Workshop on Social Intelligence in Humans and Robots at RSS 2023*, 2023.

E. L. Post. 일반 조합 결정 문제의 형식적 축소. *American Journal of Mathematics*, 65(2):197 – 215, 1943.

A. Pritzel, B. Uria, S. Srinivasan, A. P. Badia, O. Vinyals, D. Hassabis, D. Wierstra 및 C. Blundell. 신경 에피소드 제어. *International conference on machine learning*, 페이지 2827 – 2836, 2017.

M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 201 4.

C. Qian, X. Cong, C. Yang, W. Chen, Y. Su, J. Xu, Z. Liu, and M. Sun. 소프트웨어 개발을 위한 커뮤니케이션 에 이전트. *arXiv preprint arXiv:2307.07924*, 2023.

Y. Qin, S. Liang, Y. Ye, K. Zhu, L. Yan, Y. Lu, Y. Lin, X. Cong, X. Tang, B. Qian 등 Toolllm: 16000+ 실제 API를 마스터하기 위해 대규모 언어 모델을 촉진합니다. *arXiv preprint arXiv:2307.16789*, 2023.

M. Quigley. Ros: 오픈소스 로봇 운영 체제. In *IEEE International Conference on Robotics and Automation*, 20 09. URL `https://api.semanticscholar.org/CorpusID:6324125`. A. Radford, J. Wu, R. Child, D. Luan, D. A modei, I. Sutskever, et al. 언어 모델은 비지도 멀티태스크 학습기입니다. *OpenAI blog*, 1(8):9, 2019.

A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, Z. Xu, et al. 도움을 요청 하는 로봇: 대규모 언어 모델 플래너를 위한 불확실성 정렬. *7th Annual Conference on Robot Learning*, 2023.

O. J. Romero, J. Zimmerman, A. Steinfeld, A. Tomasic. 강력한 AI를 위한 대규모 언어 모델과 인지 아키텍처의 시너지 통합: 탐색적 분석. *arXiv preprint arXiv:2308.09830*, 2023.

B. Rozière, J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. Tan, Y. Adi, J. Liu, T. Remez, J. Rapin, A. Kozhevnikov, I. Evtimov, J. Bitton, M. P. Bhatt, C. C. Ferrer, A. Grattafiori, W. Xiong, A. D'efossez, J. Copet, F. Azhar, H. Touvron, L. Martin, N. Usunier, T. Scialom 및 G. Synnaeve. Code Lama: 코드를 위한 개방형 기반 모델입니다. *ArXiv*, AB S/2308.12950, 2023.

O. Rubin, J. Herzig, 및 J. Berant. 맥락 내 학습을 위한 프롬프트 검색 학습. *arXiv preprint arXiv:2112.08633*, 2 021.

E. Russek, D. Acosta-Kane, B. van Opheusden, M. G. Mattar, and T. Griffiths. 온라인 체스에서 생각하는 데 소요 된 시간은 계산의 가치를 반영합니다. *PsyArXiv*, 2022.

S. Russell 및 P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education Limited London, 2013.

D. Sadigh, A. D. Dragan, S. Sastry, S. A. Seshia. 보상 함수의 능동적 선호도 기반 학습. N. M. Amato, S. S. Sriniv asa, N. Ayanian, S. Kuindersma 편집자, *Robotics: Science and Systems XIII*, 2017.

W. Saunders, C. Yeh, J. Wu, S. Bills, L. Ouyang, J. Ward, and J. Leike. 인간 평가자를 지원하기 위한 자기 비판 모델. *arXiv preprint arXiv:2206.05802*, 2022. T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zett lemoyer, N. Cancedda, and T. Scialom. 도구 형성자: 언어 모델은 도구 사용을 스스로 학습할 수 있다. *arXiv preprint arXiv:2302.04761*, 2023. D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Ch audhary, and M. Young. 기계 학습: 기술 부채의 고금리 신용 카드. *SE4ML: Software Engineering for Machine Learning (NIPS 2014 Workshop)*에서, 2014.

T. Shi, A. Karpathy, L. Fan, J. Hernandez, P. Liang. World of Bits: 웹 기반 에이전트를 위한 오픈 도메인 플랫폼. *International Conference on Machine Learning*, 3135-3144페이지, 2017년.

N. Shinn, F. Cassano, B. Labash, A. Gopinath, K. Narasimhan, and S. Yao. Reflexion: 언어 강화 학습을 통한 언어 에이전트. *arXiv preprint arXiv:2303.11366*, 2023. M. Shridhar, X. Yuan, M.-A. Côté, Y. Bisk, A. Trischler, and M. Hausknecht. Alfworld: 대화형 학습을 위한 텍스트와 체화된 환경 정렬. *arXiv preprint arXiv:2010.03768*, 2020.

T. Silver, V. Hariprasad, R. S. Shuttleworth, N. Kumar, T. Lozano-Pérez, L. P. Kaelbling. 사전 학습된 대규모 언어 모델을 사용한 Pddl 계획. *NeurIPS 2022 Foundation Models for Decision Making Workshop*에서, 2022.

T. Silver, S. Dan, K. Srinivas, J. B. Tenenbaum, L. P. Kaelbling, M. Katz. 사전 학습된 대규모 언어 모델을 사용한 PDDL 도메인의 일반화된 계획. *arXiv preprint arXiv:2305.11014*, 2023.

I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, A. Garg. Progprompt: 대규모 언어 모델을 사용하여 상황에 맞는 로봇 작업 계획 생성. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 11523 – 11530페이지, 2023.

T. Sumers, R. Hawkins, M. K. Ho, T. Griffiths, D. Hadfield-Menell. AI가 학습할 수 있도록 말하는 방법: 지시, 설명 및 자율성. *Advances in Neural Information Processing Systems*, 35:34762 – 34775, 2022.

T. Sumers, K. Marino, A. Ahuja, R. Fergus, I. Dasgupta. 인터넷 규모의 시각 언어 모델을 구체화된 에이전트로 정제. *Proceedings of the 40th International Conference on Machine Learning*, 32797 – 32818페이지, 2023.

T. R. Sumers, M. K. Ho, R. D. Hawkins, K. Narasimhan, T. L. Griffiths. 언어적 피드백으로부터의 학습 보상. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35권, 6002~6010쪽, 2021년.

R. Sun. 인지 아키텍처를 위한 Desiderata. *Philosophical Psychology*, 17(3):341 – 373, 2004.

R. S. Sutton 및 A. G. Barto. *Reinforcement learning: An introduction*. MIT 출판부, 2018.

O. Tafjord, B. Dalvi, 및 P. Clark. 교정자: 자연어에 대한 함의, 증명 및 귀납적 진술 생성. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 3621-3634쪽, 2021.

R. Tamari, C. Shani, T. Hope, M. R. L. Petruck, O. Abend, D. Shahaf. 언어 (재)모델링: 체화된 언어 이해를 향하여. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6268 – 6281페이지, 온라인, 2020년 7월. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.559.

M. Tambe, W. L. Johnson, R. M. Jones, F. Koss, J. E. Laird, P. S. Rosenbloom 및 K. Schwamb. 대화형 시뮬레이션 환경을 위한 지능형 에이전트. *AI magazine*, 16(1):15 – 15, 1995.

M. Tang, S. Yao, J. Yang, and K. Narasimhan. 제로샷 정보 검색을 위한 추천 증강, 2023a.

Q. Tang, Z. Deng, H. Lin, X. Han, Q. Liang, and L. Sun. ToolAlpaca: Generalized Tool Learning for Language Mod els with 3000 Simulated Cases. *arXiv preprint arXiv:2306.05301*, 2023b. G. Team, R. Anil, S. Borgeaud, Y. Wu, J .-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al. Gemini: a family of high capable multimoda l models. *arXiv preprint arXiv:2312.11805*, 2023. S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Te ller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 25권, 1507~1514쪽, 2011. J. Thomason, M. Murr ay, M. Cakmak, L. Zettlemoyer. 비전과 대화 탐색. *Conference on Robot Learning*, 394~406쪽. PMLR, 2020.

A. M. Turing et al. 계산 가능한 숫자에 관하여, 결정 문제에 대한 응용. *J. of Math*, 58(345-363):5, 1936.

J. Tuyls, S. Yao, S. Kakade, 및 K. Narasimhan. 텍스트 게임에서 전략적 탐색을 위한 다단계 에피소드 제어. *arXiv preprint arXiv:2201.01251*, 2022.

K. Valmeekam, A. Olmo, S. Sreedharan, S. Kambhampati. 대규모 언어 모델은 여전히 계획할 수 없음(계획 및 변화에 대한 추론에 대한 llms의 벤치마크). *arXiv preprint arXiv:2206.10498*, 2022.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. 주의가 필요한 전부입니다. *Advances in Neural Information Processing Systems*, 30, 2017.

G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar. Voyager: 대규모 언어 모델을 갖춘 개방형 체현 에이전트. *arXiv preprint arXiv:2305.16291*, 2023a. L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J.-R. Wen. 대규모 언어 모델 기반 자율 에이전트에 대한 조사, 2023b. L. Wang, N. Yang, and F. Wei. Query2doc: 대규모 언어 모델을 사용한 쿼리 확장. *arXiv preprint arXiv:2303.07678*, 2023c. R. Wang, P. Jansen, M.-A. Côté, and P. Ammanabrolu. Sciencewo rld: 당신의 에이전트는 5학년생보다 똑똑할까요? *arXiv preprint arXiv:2203.07540*, 2022a. S. I. Wang, P. Lian g, 및 C. D. Manning. 상호작용을 통한 언어 게임 학습. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2368-23 78쪽, 2016.

X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, and D. Zhou. 자기 일관성은 언어 모델에서 사고의 사슬 추론을 개선합니다. *arXiv preprint arXiv:2203.11171*, 2022b.

J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H . Chi, T. Hashimoto, O. Vinyals , P. Liang, J. Dean 및 W. Fedus. 대규모 언어 모델의 새로운 능력. *Transactions on Machine Learning Research*, 2022a. ISSN 2835-8856. 설문조사 인증.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, D. Zhou. 사고의 사슬을 촉진하는 것은 대규모 언어 모델에서 추론을 이끌어냅니다. *arXiv preprint arXiv:2201.11903*, 2022b.

L. Weng. Llm 기반 자율 에이전트. *lilianweng.github.io*, 2023년 6월. URL `https://lilianweng.github.io/posts/2023-06-23-agent/`. J. Weston, S. Chopra, A. Bordes. 메모리 네트워크. *arXiv preprint arXiv:1410.3916*, 2014년. A. N. Whitehead와 B. Russell. *Principia mathematica to* 56*, 2권. 케임브리지 대학교 출판부, 1997년. D. E. Wilkins. *Practical planning: extending the classical AI planning paradigm*. Elsevier, 2014년. T. Winograd. 자연어 이해. *Cognitive psychology*, 3(1):1 – 191, 1972년. L. Wong, G. Grand, A. K. Lew, N. D. Goodman, V. K. Mansinghka, J. Andreas, J. B. Tenenbaum. 단어 모델에서 세계 모델로: 자연어에서 확률적 사고 언어로 번역. *arXiv preprint arXiv:2306.12672*, 2023년.

R. E. Wray, J. R. Kirk, J. E. Laird, et al. 인지 에이전트를 위한 지식 소스로서의 언어 모델. *arXiv preprint arXiv:2109.08270*, 2021.

Q. Wu, G. Bansal, J. Zhang, Y. Wu, S. Zhang, E. Zhu, B. Li, L. Jiang, X. Zhang, and C. Wang. Autogen: En- abling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023. T. Wu, E. Jiang, A. Donsbach, J. Gray, A. Molina, M. Terry, and C. J. Cai. Promptchainer: Chaining large language model pr ompts through visual programming. *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, 1 – 10페이지, 2022a. T. Wu, M. Terry, and C. J. Cai. AI chains: Chainin g large language model prompts를 통한 투명하고 제어 가능한 인간-AI 상호작용. *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 1 – 22페이지, 2022b.

Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou 등. 대규모 언어 모델 기반 에이전트의 부상과 잠재력: 조사. *arXiv preprint arXiv:2309.07864*, 2023. Y. Xia, M. Shenoy, N. Jazdi, and M. Weyrich. 자율 시스템을 향하여: 대규모 언어 모델 에이전트로 강화된 유연한 모듈식 생산 시스템. *arXiv preprint arXiv:2304.14721*, 2023. Y. Xie, T. Xie, M. Lin, W. Wei, C. Li, B. Kong, L. Chen, C. Zhuo, B. Hu, and Z. Li. Olagpt: 인간과 같은 문제 해결 능력으로 LLM 강화. *arXiv preprint arXiv:2305.16334*, 2023.

B. Xu, X. Liu, H. Shen, Z. Han, Y. Li, M. Yue, Z. Peng, Y. Liu, Z. Yao 및 D. Xu. Gentopia: 도구로 강화된 LLM을 위한 협업 플랫폼입니다. *arXiv preprint arXiv:2308.04030*, 2023a.

B. Xu, Z. Peng, B. Lei, S. Mukherjee, Y. Liu, D. Xu. Rewoo: 효율적인 증강 언어 모델을 위한 추론과 관찰 분리. *arXiv preprint arXiv:2305.18323*, 2023b. B. Xu, A. Yang, J. Lin, Q. Wang, C. Zhou, Y. Zhang, Z. Mao. ExpertPro mpting: 대규모 언어 모델에 뛰어난 전문가가 되도록 지시. *arXiv preprint arXiv:2305.14688*, 2023c. J. Yang, A. Prabhakar, K. Narasimhan, S. Yao. Intercode: 실행 피드백을 통한 대화형 코딩 표준화 및 벤치마킹. *arXiv preprint arXiv:2306.14898*, 2023. S. Yao 및 K. Narasimhan. 디지털 세계의 언어 에이전트: 기회와 위험. *princeton- nlp.github.io*, 2023년 7월. URL `https://princeton-nlp.github.io/language-agent-impact/`. S. Yao, R. Rao, M. Hausknecht, and K. Narasimhan. Keep CALM and explore: 텍스트 기반 게임에서 액션 생성을 위한 언어 모델. *arXiv preprint arXiv:2010.02903*, 2020. S. Yao, H. Chen, J. Yang, and K. Narasimhan. 웹쇼: 기반 언어 에이전트를 통한 확장 가능한 실제 세계 웹 상호 작용을 향해. *Advances in Neural Information Processing Systems*, 35:20744 – 20757, 2022a.

S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. React: 언어 모델에서 추론과 행동의 시너지 효과. *arXiv preprint arXiv:2210.03629*, 2022b.

S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan. 생각의 나무: 대규모 언어 모델을 사용한 의도적인 문제 해결. *arXiv preprint arXiv:2305.10601*, 2023.

E. Zelikman, Y. Wu, J. Mu, and N. Goodman. STaR: 추론을 추론으로 부트스트래핑. *Advances in Neural Information Processing Systems*, 35:15476 – 15488, 2022.

A. Zeng, M. Attarian, B. Ichter, K. Choromanski, A. Wong, S. Welker, F. Tombari, A. Purohit, M. Ryoo, V. Sindhwani, et al. 소크라테스 모델: 언어로 제로샷 멀티모달 추론 구성하기. *arXiv preprint arXiv:2204.00598*, 2022.

C. Zhang, L. Wong, G. Grand, J. Tenenbaum. 확률적 프로그램과 시뮬레이션된 세계를 통한 근거 있는 물리적 언어 이해. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 45권, 2023a.

T. Zhang, F. Liu, J. Wong, P. Abbeel, J. E. Gonzalez. 회고의 지혜는 언어 모델을 더 나은 지시 추종자로 만든다. *arXiv preprint arXiv:2302.05206*, 2023b.

Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, W. B. Dolan. Dialogpt: 대화 응답 생성을 위한 대규모 생성 사전 훈련. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 270~278페이지, 2020. W. J. Zhao, R. Richie, S. Bhatia. 기억으로부터의 결정의 프로세스와 내용. *Psychological Review*, 129(1):73, 2022. V. Zhong, A. W. Hanjie, S. Wang, K. Narasimhan, L. Zettlemoyer. SILG: 다중 도메인 심볼릭 상호작용 언어 기초 벤치마크. *Advances in Neural Information Processing Systems*, 34: 21505~21519, 2021.

C. Y. Zhou, D. Talmi, N. Daw, and M. G. Mattar. 순차적 의사 결정 작업에서 모델 기반 평가를 위한 에피소드 검색, 2023a. H. Zhou, M. Huang, T. Zhang, X. Zhu, and B. Liu. 감정 채팅 머신: 내부 및 외부 메모리를 사용한 감정 대화 생성. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32권, 2018년. S. Zhou, U. Alon, F. F. Xu, Z. Jiang, and G. Neubig. 문서 생성: 문서를 검색하여 코드 생성.
The Eleventh International Conference on Learning Representations, 2022a. S. Zhou, F. F. Xu, H. Zhu, X. Zhou, R. Lo, A. Sridhar, X. Cheng, Y. Bisk, D. Fried, U. Alon, et al. WebArena: 자율 에이전트를 구축하기 위한 현실적인 웹 환경. *arXiv preprint arXiv:2307.13854*, 2023b. Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, J. Ba. 대규모 언어 모델은 인간 수준의 프롬프트 엔지니어입니다. *arXiv preprint arXiv:2211.01910*, 2022b.