도구 증강 에이전트: 자율성에서 상호작용으로의 진화

조요한, 이종근
서울대학교 데이터사이언스대학원
yohan.jo@snu.ac.kr

Tool-Augmented Agents: Evolution from Autonomy to Interaction

Yohan Jo, Jonggeun Lee
Graduate School of Data Science, Seoul National University

요 약

도구 증강 에이전트(Tool-Augmented Agents)는 인공지능 모델에 외부 도구를 결합하여 추론 능력과 자율성을 확장하는 방향으로 활발히 연구되어 왔다. 특히 에이전트의 기반이 되는 거대언어모델이 급격하게 발전함에 따라, 많은 초기 연구들은 에이전트의 계획 능력 및 오류로부터 복구하는 능력 등 에이전트의 인지적 능력과 자율성에 주로 초점을 맞추었다. 그러나 최근 들어 에이전트 연구의 관심은 인간과의 상호작용으로 확장되고 있다. 예를 들어, 필요한 정보를 얻어내기 위해 사용자에게 적절한 질문을 던지거나 사용자의 선호에 맞춰 개인화하는 등의 연구가 진행 중이다. 본 기고문은 이러한 흐름을 종합적으로 검토하며 도구 증강 에이전트 연구가 자율성 중심에서 인간 중심의 상호작용으로 진화해 온 과정을 조망한다. 나아가 앞으로의 연구 과제를 제안하며 도구 에이전트의 미래 방향을 모색한다. 이를 통해, 본 기고문이 에이전트 연구자들에게 하나의 지도 역할을 하고 함께 토론하기 위한 공통 재료가 되기를 희망한다.

1. 서론

단순한 대화 상대를 넘어 스스로 업무를 처리하는 인공지능 에이전트가 우리 일상과 산업의 지형을 바꾸고 있다. 고객 응대를 완전히 자동화하는 것에서부터, 복잡한 코드를 작성하고 테스트하는 소프트웨어 개발, 음성을 통한 스마트 기기 제어, 신약 후보 물질을 탐색하는 과학 연구에 이르기까지, 에이전트는 다양한 분야에서 인간의 능력을 증강시키는 능동적인 파트너로 자리매김하고 있다.

이러한 변화의 중심에는 거대언어모델(Large Language Model)이 있다. 언어모델은 다양한 자연어 처리 과업에서 인간에 필적하는 성능을 보이며 인공지능 분야에 혁신을 가져왔다. 그러나 일반적인 언어모델은 학습된 데이터에 제한되어 실시간 정보에 접근하거나 외부 세계와 직접 상호작용할 수 없는 본질적인 한계를 지닌다. 이러한 한계를 극복하고 인공지능의 활용 범위를 현실 세계로 확장하려는 시도 속에서, 외부 도구와 상호작용하는 **도구 증강 에이전트(Tool-Augmented Agents)** 또는 짧게 **도구 에이전트**가 핵심적인 연구 분야로 부상했다.

인공지능 분야에서 '에이전트'란 특정 목표 달성을 위해 자율적으로 판단하고 행동하는 소프트웨어를 말하는데, 이는 자신의 환경을 인지하고 목표를 위해 행동하는 주체라는 인공지능의 고전적 개념에 뿌리를 둔다 [1, 2]. 오늘날의 도구 에이전트는 바로 이 개념을 언어모델에 접목한 것으로, 언어모델에 외부 도구를 사용하는 능력을 부여하여 단순히 정보를 처리하는 수동적 모델을 넘어 실제 과업을 수행하는 능동적 행위자로 거듭나게 한다. 특히 이 기고문에서는 **API(Application Programming Interface) 호출**을 통해 외부 도구와 소통하는 에이전트에 초점을 맞춘다. 이는 인공지능 연구의 패러다임을 정보 '처리'에서 '행동'으로 전환하는 중요한 전환점이며, 인공지능이 가상 세계를 넘어 현실과 상호작용하는 구체적인 실현 방안을 제시한다는 점에서 큰 의의를 갖는다.

이러한 에이전트 연구의 초기 흐름은 에이전트라는 이름의 근본적인 의미인 **자율성**을 향상시키는 데 주로 초점을 맞추었다(2장에서 자세히 소개). 자율성에는 에이전트의 다양한 인지적 능력이 포함되는데, 크게 계획 능력, 추론 능력, 물리적 이해 능력, 도구 선택 능력, 경험 활용 능력, 오류 복구 능력, 안정성, 효율성 등이 있다. 그러나 에이전트의 자율성이 아무리 향상되어도 결국 에이전트를 사용하는 주체는 인간이므로, 최근 들어 연구의 관심은 **인간과의 상호작용**으로 확장되고 있다(3장에서 자세히 소개). 여기에 필요한 주요 능력에는 대화 상태를 정확히 추적하는 능력, 사용자에게 질문을 통해 필요한 정보를 선제적으로 수집하는 능력, 비협조적인 사용자에 대처하는 능력, 그리고 사용자의 선호에 개인화하는 능력이 포함된다.

본 기고문은 이러한 흐름을 종합적으로 검토하며 도구 에이전트 연구가 자율성 중심(2장)에서 인간 중심의 상호작용(3장)으로 진화해 온 과정을 조망한다. 나아가 이 전환이 갖는 의의와 앞으로 남은 연구 주제들을 언급하며 끝내고자 한다(4장). 이를 통해, 도구 에이전트 연구에 입문하는 연구자들에게 하나의 지도가 되기를 바라고, 이미 연구를 진행 중인 연구자들에게는 함께 토론을 발전시키기 위한 유용한 공통 재료가 될 수 있기를 바란다.

2. 자율성 중심의 에이전트 연구

언어모델에 기반한 도구 에이전트 연구가 본격화 되면서, 초반부터 지금까지 에이전트의 자율성 향상을 위한 연구들이 꾸준히 진행되고 있다. 이와 관련된 능력은 크게 8가지로 구분해 볼 수

자율성	
추론 능력	토큰 생성을 통해 생각하는 과정을 거치면서 현 상황을 분석하고 수행해야 할 일을 결정하는 능력
계획 능력	복잡한 과업이 주어졌을 때, 이를 풀기 위한 작은 단계들로 분해하여 구체적인 계획을 세우는 능력
물리적 이해 능력	도구가 사용되는 물리적인 환경(시간성 및 다양한 제약들)에 대한 이해 능력
도구 선택 능력	상황에 적절한 도구를 검색하거나 도구 리스트에서 적절한 도구를 선택하는 능력
경험 활용 능력	수행을 통해 얻은 성공 및 실패 경험들을 활용하는 능력
오류 복구 능력	과업 수행 중에 오류를 맞닥뜨렸을 때, 이를 인지하고 정상적인 수행 궤도로 복구하는 능력
안전성	외부 도구와 상호작용하는 과정에서 위험한 행위를 식별하고 피하는 능력
효율성	작업을 저비용으로 수행하는 능력
인간과의 상호작용	
대화 상태 추적 능력	사용자의 의도와 요구사항을 동적으로 업데이트하고 추적하여, 대화 맥락에 맞게 사용자의 요구사항을 처리할 수 있는 능력
선제적 정보 수집 능력	사용자의 요구나 과업 수행에 필요한 정보가 불완전할 때 선제적으로 질문하여 명확화하는 능력
비협조적 사용자 대처 능력	협조적이지 않고 다루기 어려운 사용자의 행동에 대처하는 능력
개인화 능력	사용자 개인의 선호에 맞춤화된 제안 및 선제적인 제안을 하는 능력

표 1: 도구 에이전트의 핵심 능력

있다: 추론 능력, 계획 능력, 물리적 이해 능력, 도구 선택 능력, 경험 활용 능력, 오류 복구 능력, 안정성, 효율성. 각 능력에 대한 간단한 정의는 표 1에 요약되어 있다. 앞으로 기술할 서브섹션별로 각 능력에 관한 연구들을 소개하도록 한다.

2.1 추론 능력

에이전트의 추론 능력이란 토큰 생성을 통해 생각하는 과정을 거치면서 현 상황을 분석하고 앞으로 수행해야 할 일을 결정하는 일반적인 능력을 가리킨다. 단순한 질의응답 상황에서는 에이전트가 Chain of Thought(CoT) [3]을 통해 사용자의 질문 또는 요청을 분석하고 적절한 응답에 단계적으로 도달해 나갈 수 있다. 그러나 도구를 활용해야 하는 어려운 과업에서는 CoT를 통해 모델의 파라미터에 내재된 지식만 사용해서는 사용자의 요청을 해결하기 어렵고, 외부의 도구와 상호작용을 하면서 단계를 진행해 나가야 한다.

이에 따라, CoT를 확장한 ReAct [4]와 ART [5] 같은 방법들이 등장했다. CoT가 생각들의 사슬로만 구성되어 있는 반면, ReAct와 ART는 각 생각 이후에 이에 근거한 API 호출 및 결과 관찰이 추가된다. 가령, 사용자가 "내 은행계좌에 있는 돈을 다 모으면 쿠팡에서 어떤 노트북을 살 수 있어?"라고 질문하면, 에이전트는 첫 번째 단계로 "사용자의 모든 은행계좌 잔액을 확인해야겠다"고 판단한 뒤, 은행 API를 호출하여 계좌 목록과 각 잔액을 조회한다. 여러 계좌(예: 국민은행 20만원, 카카오뱅크 30만원, 토스뱅크 15만원)가 확인되면, 총 65만원이 사용 가능함을 파악하고, 다음 단계로 "65만원 이하로 구매 가능한 노트북을 검색해야겠다"고 판단하여 쿠팡 API를 호출해 가격대에 맞는 노트북 목록을 받아 사용자에게 제시한다. 이처럼 에이전트는 사용자의 요청에서 필요한 정보를 자율적으로 파악하고, 여러 API를 순차적으로 호출하며, 각 단계의 결과를 다음 추론에 활용함으로써 복잡한 과업을 수행할 수 있다. 많은 에이전트들이 ReAct 방식의 추론을 뼈대로 갖고 있다.

2.2 계획 능력

에이전트의 계획 능력은 복잡한 과업이 주어졌을 때, 이를 해결 가능한 작은 단계들로 분해하고 구체적인 실행 경로를 결정하는 능력을 의미한다. 이 과정에서 에이전트는 하위 과업들 간의 논리적 순서와 의존 관계를 파악하고, 사용 가능한 도구나 API를 적절히 선택하여 목표를 달성해야 한다. 특히 여러 API 호출이 필요한 복잡한 과업에서는 API들 간의 의존성을 이해하고 적절한 호출 순서를 결정하는 것이 중요하다.

전통적인 Plan-and-Execute [6] 방식은 과업 수행 전에 전체 계획을 구체적으로 수립한 후 순차적으로 실행한다. 반면 ADaPT [7]는 초기에 상대적으로 거시적인 수준의 계획을 수립하고, 실행 과정에서 특정 하위 과업이 실패할 경우 해당 부분만을 더 작은 단위로 재분해하여 계획을 구체화한다. 이는 에이전트가 자신의 실행 능력과 과업의 복잡도를 바탕으로 계획의 세밀도를 동적으로 조절하게 하여, 단순한 과업에서는 불필요한 계획 비용을 줄이고 복잡한 부분에만 집중함으로써 효율성을 높인다.

한편, 기존의 ReAct 방식이 단일 경로만을 순차적으로 따라가는 것과 달리, DFSDT [8] 방식은 깊이 우선 탐색 기반의 의사결정 트리(Depth-First Search-based Decision Tree)를 통해 여러 가능한 실행 경로를 탐색하고 평가한다. 한 경로를 깊이 탐색하다가 실패하거나 막다른 상태에 도달하면 이전 분기점으로 돌아가 다른 유망한 경로를 시도하는 방식으로, 에이전트는 하나의 시도가 실패하더라도 대안적인 해결책을 찾아낼 수 있다. 이를 통해 단일 경로 추적의 한계를 극복하고 보다 강건하고 유연한 계획

수립이 가능하다.

또한, 대부분의 계획 방법이 초기 상태에서 시작하여 목표를 향해 순차적으로 계획을 세우는 것과 달리, Reverse Chain [9]은 최종적으로 호출해야 할 API를 먼저 결정하고 그 API에 필요한 인자들을 채워나가는 역방향 방식으로 동작한다. 만약 특정 인자를 채우기 위해 다른 API의 출력이 필요하다면, 해당 API를 호출하는 것을 새로운 하위 목표로 설정하고 이 과정을 재귀적으로 반복한다. 이 방식은 최종 목표에서 벗어날 위험을 줄여 계획의 안정성과 제어 가능성을 높이는 장점이 있다.

2.3 물리적 이해 능력

도구 에이전트가 상용화에 성공한 최초의 영역들 중 하나가 바로 물리적 도구와 결합된 스마트홈 분야이다. 아마존의 Alexa, 구글의 Google Home 등이 대표적인 예이다. 이런 분야에서는 에이전트가 단순히 도구의 기능을 아는 것을 넘어 물리적 이해 능력, 가령 어떤 명령을 수행하는 데 걸리는 시간으로 인한 시간적 제약 및 기계의 특정 상태에서만 어떤 기능이 작동하는 것과 같은 물리적 제약을 이해하는 것이 중요하다.

그러나 많은 상용 스마트홈 에이전트들이 이런 능력으로부터 한참 못미쳐 있는데, 이는 스마트홈 에이전트에 관한 연구들이 있긴 하지만 현실에서 필요로 하는 물리적 이해 능력의 복잡도를 충분히 반영하지 못하고 정적인 데이터셋에 머무는 한계 때문이다 [10, 11]. 이를 극복하기 위해, SimuHome [12]은 실제 스마트홈 산업 표준인 Matter[1] 프로토콜을 기반으로 구축된 가상의 스마트홈 시뮬레이션 환경을 제공한다. SimuHome 내에서는 다양한 스마트 기기들이 Matter 기반 API를 통해 작동 가능하며, 시간의 흐름에 따른 환경 변화(예: 에어컨을 켜면 서서히 온도가 내려감)와 기기의 물리적 제약들(예: 세탁기가 작동 중일 때는 문을 열 수 없음)을 현실적으로 모델링한다. 이를 통해 에이전트는 가상의 스마트 기기들과 상호작용을 하며 학습되거나 평가될 수 있고, 이러한 에이전트는 현실의 Matter 기반 스마트 기기에 바로 결합하여 사용할 수 있다. 또한 함께 제공되는 벤치마크는 복잡한 물리적, 시간적 제약 상황에서 에이전트의 문제 해결 능력을 다각도로 평가할 수 있게 해준다.

2.4 도구 선택 능력

에이전트는 가용한 많은 도구들 중에서 현재 상황에 적절한 도구를 검색하거나, 검색을 통해 제공된 리스트들 중에서 적절한 도구를 선택하는 능력이 필요하다. 이 장에서는 먼저 도구 검색 기술이 3단계로 진화해 온 과정에 대해 소개하고, 그 후에는 검색 기술이 아닌 도구 정보 자체를 가공하여 에이전트의 활용성을 높이는 연구에 대해 소개한다.

1) https://csa-iot.org/all-solutions/matter/

2.4.1 도구 검색의 진화

전통적 검색 초기 단계에서는 기존 정보 검색 기술을 도구 검색 문제에 직접 적용했다. 즉, 사용자의 지시는 검색 질의로 사용되고, 도구 설명 문서(Documentation)는 검색 대상 문서로 취급된다 [8]. 하지만 최근 연구에 따르면, 전통적인 검색 기술은 문서의 주제적 유사성을 찾는 데 최적화되어 있기 때문에 도구의 기능과 활용 상황을 이해해야 하는 도구 검색에서는 지시와 도구 문서 간 근본적인 의미적 불일치가 발생한다. 이로 인해 사용자의 일상 언어와 기술적인 API 설명 간의 큰 어휘 차이를 극복하지 못하고, 결국 우수한 검색 모델조차도 저조한 성능을 보이게 된다 [13].

에이전트 활용 검색 위 한계를 개선하기 위해 등장한 연구들에서는 에이전트의 강력한 추론 능력을 검색 과정에 직접 활용하는 것이 특징이다. 도구를 검색하기 전에 에이전트를 사용하여 사용자의 모호한 질의를 더 명확한 질의로 재작성하거나 [14], 에이전트가 검색기 모델에 피드백을 제공하여 두 모델 간의 이해도 격차를 줄이고 검색을 점진적으로 개선하기도 한다 [15].

생성형 검색 최근에는 검색과 생성 사이의 경계를 허물며, 도구 선택을 검색 문제가 아닌 다음 토큰 예측 작업으로 재구성한다 [16, 17]. 전체 도구 세트에 대한 지식이 에이전트 언어모델의 매개변수에 직접 통합되기 때문에, 에이전트가 일반적으로 텍스트를 생성하는 과정에서, 필요한 도구의 이름이나 식별자를 자연스럽게 생성할 수 있게 한다. 이를 통해 별도의 검색 모듈이 더 이상 필요 없는 깊이 통합되고 효율적인 시스템이 만들어진다.

2.4.2 도구 정보 가공

검색 과정에 관여하는 모델들을 개선하기 보다는 도구에 관한 정보 자체를 가공하여 에이전트가 효과적으로 활용할 수 있는 형태로 제공하는 시도들도 있어 왔다. 이 방법들은 에이전트나 검색 관련 모델들을 도구에 맞춰 학습하거나 복잡한 과정을 거칠 필요가 없어서 상대적으로 비용이 적다.

초기 연구들은 API 문서를 언어모델이 더 잘 이해할 수 있는 형태로 재구성하는 데 초점을 맞추었다. EasyTool [18]은 복잡하고 장황한 API 문서를 간결하고 구조화된 형식으로 정제할 뿐만 아니라, 각 API를 실제로 사용하는 구체적인 시나리오를 생성하여 함께 제공한다. 이러한 사용 예시를 통해 언어모델은 API의 기능을 추상적인 설명이 아닌 실제 활용 맥락에서 이해할 수 있게 된다.

그러나 앞서 계획 능력에서 언급하였듯이, API들 간의 관계를 이해하고 호출 순서를 적절히 결정하는 것이 중요함에도, 단순한 문서 정제 및 예시 제공만으로는 API들 간의 의존 관계와 같은 구조적 정보를 명시적으로 제공하기 어렵다는 한계가 있다. 이를 해결하기 위해 In-N-Out [19]은 API 문서들을 API들 간의 의

존 그래프로 변환하여, 어떤 API의 출력이 다른 API의 입력으로 사용될 수 있는지를 구조적으로 표현한다. 이를 통해 에이전트는 복잡한 다중 API 과업에서 호출 순서를 더 효과적으로 결정할 수 있게 된다.

비록 문서를 정제하고 구조적 정보를 제공하더라도, 에이전트가 실제로 API를 호출할 때 정확한 API 이름과 인자명을 생성하는 것은 여전히 어려운 문제로 남는다. 특히 소형 언어모델 기반 에이전트의 경우, 사전학습 과정에서 익숙해진 명명 규칙과 실제 API 및 인자들의 이름이 다를 때 환각 현상이 발생하기 쉽다. PA-Tool [20]은 에이전트를 도구에 맞추는 대신, 도구 스키마를 에이전트에 맞춰 수정하는 역발상적 접근을 제안한다. 데이터 오염 연구, 즉 어떤 언어모델이 사전학습 과정에서 특정 텍스트에 얼마나 노출되었는지 분석하는 방법을 사용해, API와 인자들의 이름을 해당 에이전트에게 익숙한 이름으로 교체한다. 이를 통해 일체의 학습 과정 없이, 그리고 특정 도구들에 과적합되거나 다른 도구들을 잊어버릴 우려 없이 소형 에이전트의 도구 호출 정확도를 크게 향상시킨다.

2.5 경험 활용 능력

에이전트가 경험을 얻는 방법은 크게 두 가지이다. 첫째, 학습의 관점에서 에이전트가 많은 경로를 탐색하면서 얻은 보상 및 손해 정보를 학습함으로써, 경험이 에이전트의 파라미터에 내재될 수 있다. 둘째, 에이전트가 과업을 수행하며 경험하는 성공과 실패 상황이 외부 메모리에 축적 및 가공되고, 에이전트는 추후 실행 시 관련된 경험 정보를 가져와 활용할 수 있다.

2.5.1 경험 내재화

강화학습은 에이전트가 환경과의 상호작용 경험을 파라미터에 내재화하는 대표적인 방법이다. 또한 이렇게 수집된 상호작용 데이터를 사용해 에이전트를 지도학습 함으로써 경험을 내재화 시킬 수도 있다.

ToolRL [21]은 도구 호출의 복잡한 구조를 반영한 세밀한 보상 체계를 제안한다. 단순히 최종 답변의 정확성만으로 보상을 주는 것이 아니라, 출력 형식의 올바름, 도구 이름의 정확성, 파라미터 이름의 정확성, 파라미터 값의 정확성을 각각 평가하여 보상을 부여한다. 이렇게 세분화된 보상 신호는 에이전트가 다단계 도구 호출 과정에서 어떤 부분이 올바르고 어떤 부분이 잘못되었는지 명확히 학습할 수 있게 하며, 이러한 경험이 누적되어 파라미터에 내재화됨으로써 새로운 도구나 시나리오에도 일반화할 수 있게 된다.

LOOP [22]는 에이전트를 AppWorld [23]와 같은 API 시뮬레이션 환경에서 직접 훈련시키는 강화학습 방법이다. 에이전트는 환경과 상호작용하며 보상을 받고, 수집된 경험 데이터를 여러 차례의 파라미터 업데이트에 재사용하여 효율적으로 학습한다. 이 과정에서 에이전트는 단순히 과업 해결법을 암기하는 것을 넘어, 불확실할 때 API 문서를 먼저 참고하거나 근거 없는 추측을 피하는 등 일반화되고 안정적인 문제 해결 전략을 스스로 터득하게 된다.

ETO [24]와 Agent Q [25]는 에이전트가 환경을 탐색하며 수집한 성공 및 실패 경로를 활용하여 학습한다. 먼저 다양한 경로를 시도하며 어떤 경로가 과업 완수에 성공하고 어떤 경로가 실패하는지 기록한 뒤, Direct Preference Optimization [26]을 통해 성공 경로는 선호하고 실패 경로는 회피하도록 모델 파라미터를 조정한다. DPO는 전통적인 강화학습 알고리즘을 사용하지 않고 선호도 데이터로부터 직접 학습하는 방식이므로, 이 접근법은 환경과의 실시간 상호작용이나 복잡한 보상 함수 설계 없이도 오프라인으로 효과적인 학습이 가능하다.

2.5.2 경험 외부 메모리화

에이전트가 겪은 경험을 파라미터에 내재화하지 않고 대신 외부 메모리에 명시적으로 저장하고 활용하는 접근법도 있다. 경험을 외부 메모리에 저장하고 활용하는 단순하면서도 효과적인 초기 연구로는 Reflexion [27]이 있다. 에이전트가 특정 과업 수행에 실패했을 때, 실패한 경로에 대해 스스로 '언어적 성찰'을 수행하여 외부 메모리에 오답노트처럼 기록한다. 이후 동일한 과업을 다시 시도할 때 이 성찰 기록을 참고하여 같은 실수를 반복하지 않도록 계획을 수정한다.

Reflexion이 단일 과업의 성공에 집중한다면, ExpeL [28]은 여러 과업들에서 얻은 경험을 일반화하여 처음 보는 새로운 과업을 해결하는 데 활용한다. 구체적으로 에이전트는 다양한 과업들을 수행하며 얻은 성공 및 실패 경험들을 외부 '경험 풀'에 기록한다. 이후, 이 경험 풀로부터 일반화할 수 있는 '통찰'을 자연어 형태로 추출하고(언어모델 이용), 새로운 과업이 주어졌을 때 경험 풀에서 가장 유사한 성공 사례를 검색하여 참고한다. 이처럼 통찰과 유사 경험을 함께 활용하는 방식은 에이전트 언어모델의 파라미터를 직접 수정하지 않고도 여러 과업에 걸쳐 성능을 향상시키는 접근법이다.

에이전트가 겪는 오류의 근본 원인 중 하나가 불완전하고 모호한 API 문서에 있다. DRAFT [29]는 에이전트의 실행 경험을 바탕으로 API 문서를 동적으로 개정한다. 이 프레임워크는 '탐색기', '분석기', '개정기' 세 에이전트의 협력을 통해 작동한다. 먼저 탐색기 에이전트가 현재 API 문서를 바탕으로 다양한 시나리오를 탐색하며 실행 기록을 수집한다. 그러면 분석기 에이전트가 이 기록을 검토하여 API 문서의 어떤 부분이 오류를 유발했는지, 혹은 어떤 설명이 부족했는지를 분석하여 개선 제안을 내놓는다. 마지막으로 개정기 에이전트가 이 제안을 반영하여 API 문서를 개정한다. Reflexion이나 ExpeL이 실패 경험을 별도의 메모리(오답노트, 경험 풀)에 저장하는 반면, DRAFT는 그 경험의 교훈을 API 문서에 직접 통합한다. 이렇게 개선된 문서는 이후의 모든 에이전트에게 더 나은 가이드라인을 제공하는 영구적인

외부 메모리 역할을 하게 되며, 이를 통해 근본적으로 오류 발생 가능성을 줄인다.

2.6 오류 복구 능력

오류 복구 능력이란 에이전트가 과업을 수행하는 중에 오류를 만나거나 막다른 경로임을 파악했을 때, 올바른 경로로 돌아갈 수 있는 능력을 말한다. 이전 장에서 다룬 경험 활용 능력이 오류를 근본적으로 피하기 위함이라면, 오류 복구 능력은 이미 오류를 만났을 때 대처하는 능력이라 볼 수 있다.

SCoRe [30]는 멀티턴 강화학습을 통해 에이전트가 첫 번째 시도에서 오류를 범했을 때 이를 참조하면서 두 번째 시도에서 올바르게 수정하는 능력을 학습한다. 핵심은 두 번째 시도의 보상에 첫 번째 시도 대비 개선 정도를 측정하는 추가 보상을 부여하는 것이다. 이러한 과정에서 에이전트는 처음부터 정답을 맞히는 것뿐만 아니라, 자신의 이전 응답을 되돌아보고 개선하는 전략을 학습하게 된다.

E^2CL [31]은 에이전트가 환경을 탐색하며 발생시킨 오류와 그에 대한 환경의 피드백을 수집하여 오류 복구 능력을 학습한다. 에이전트는 전문가 궤적을 따라가면서 일부러 다른 행동을 시도해보거나 자유롭게 환경을 탐색하면서 실행 불가능한 행동들을 경험한다. 오류 행동에 대한 환경의 피드백과 전문가의 올바른 행동을 연결하여 학습 데이터를 구성하고, 에이전트가 오류를 인식하고 올바른 행동으로 복구하는 과정을 학습한다.

Agent-R [32]은 과업 수행 중 실패가 시작된 지점을 파악하여 그 지점에서부터 올바른 경로로 복구하는 훈련에 초점을 맞춘다. 이 방법은 몬테카를로 트리 탐색(Monte Carlo Tree Search)을 활용하여 성공 경로와 실패 경로를 다양하게 탐색한 뒤, 실패 경로에서 오류가 발생한 첫 번째 지점과 성공 경로의 시작 지점을 식별한다. 그리고 실패 지점으로부터 성공 시작점으로 이동하는 행위가 추가된 새로운 경로를 합성한 뒤에, 이를 학습 데이터로 사용한다. 이렇게 학습된 모델은 과업 수행 중에 실패로 판단되는 지점에 도달하면 스스로 올바른 경로로 이동하는 능력을 배우게 된다.

2.7 안전성

도구 에이전트의 안정성이란 외부 API와 상호작용하는 과정에서 유해한 행동을 방지하고, 사용자가 위험한 요청을 하더라도 거절할 수 있는 능력을 의미한다. 이러한 안정성과 관련된 연구는 크게 안전성 평가 연구와 안전성 강화 연구로 구분할 수 있다.

도구 에이전트의 안전성 평가를 위한 초기 연구에서는 도구 시뮬레이션 환경을 제공하여 에이전트와 도구 간 상호작용 중 발생할 수 있는 위험한 행동을 효율적으로 탐지하는 접근이 제안되었다 [33]. 이후 연구들은 비신뢰성 데이터나 악의적인 도구 출력으로부터 비롯되는 프롬프트 공격을 중심으로, 현실적인 도구 호출 환경에서의 취약성을 검증하는 방향으로 확장되었다 [34, 35]. 최근에는 보다 포괄적인 행위 및 보안 리스크를 다루며, 유해 행동과 방어 전략을 분류하기 위한 표준화된 평가 틀을 제시하고 있다 [36, 37]. 이러한 흐름을 통해 도구 에이전트의 복합적인 안전성 문제를 체계적이고 재현 가능한 방식으로 평가하려는 방향으로 연구가 진행되고 있다.

에이전트의 안전성을 강화하려는 접근도 활발히 진행되고 있다. 최근 연구들은 안전성을 단순히 외부에서 통제하는 것이 아니라, 정책적 규칙과 판단 과정을 에이전트 내부에 통합해 스스로 안전한 결정을 내릴 수 있도록 만드는 방향으로 발전하고 있다 [38, 39]. 이는 에이전트의 잠재적인 행동을 명시적 제약과 비교 및 검증하거나, 에이전트의 계획 또는 결정 단계에서 규범적 원칙을 준수하도록 설계하는 것을 포함한다. 이처럼 외부 도구와의 상호작용 전 과정에서 스스로 안전성을 유지할 수 있는 에이전트를 구축하려는 방향으로 진화하고 있다.

2.8 효율성

에이전트가 과업을 수행하는 과정에서 종종 과도한 도구 호출이나(특히 오류로 인한 재시도 과정에서) 최적화되지 않은 도구 사용으로 인해 시간과 비용이 많이 들기 때문에, 최근에는 에이전트의 효율성이 중요한 능력으로 주목받기 시작했다. 이러한 효율성에 관한 연구는 주로 도구의 과도한 사용 방지, 도구 실행 시간 단축, 추론을 위한 과도한 토큰 사용 방지를 중심으로 이루어졌다.

가장 활발히 진행되는 주제는 에이전트가 도구를 호출해야 하는 상황과 호출 없이 처리할 수 있는 상황을 스스로 판단하는 메타인지 능력에 관한 연구이다. MetaTool [40]은 이를 평가할 수 있는 대표적인 벤치마크를 제시한다. 구체적인 방법으로, MeCo [41]는 에이전트 언어모델이 이미 내부적으로 이러한 메타인지를 어느 정도 가지고 있다는 가정 하에 모델의 내부 표상으로부터 신호를 잡아내어 도구 호출 여부를 제어하고, SMART [42]는 모델의 추론 과정을 통해 불필요한 호출을 줄이면서 정확도를 유지하거나 개선한다.

도구 실행 시간은 실제 에이전트의 작업 시간에서 큰 비중을 차지하는 병목이므로, 이를 줄이려는 연구도 활발히 진행 중이다. CATP-LLM [43]은 도구를 어떤 순서와 방식으로 사용할지를 계획할 때 비용 정보를 고려하도록 학습하는 접근을 제안한다. 한편 LLMCompiler [44]는 도구 간 호출 종속성을 분석하여 최대한 병렬적으로 도구를 호출함으로써 전체 응답 시간을 단축하는 전략을 제시한다.

최근의 많은 언어모델들이 토큰 기반의 추론(예: Chain-of-Thought)을 길게 내뱉는 추세가 되면서, 언어모델의 불필요한 과잉 추론이 효율성을 저해하는 요소로 지적되기 시작했다. 과잉 추론을 억제하는 기술들이 제안되면서, 특히 도구 에이전트의 추론 과정에 적용하는 연구도 등장했다. ThinkBrake [45]는 에이전트가 추론 과정에서 이미 적절한 도구와 인자를 예측해낸 시점

이후의 추가적인 추론을 과잉 추론이라고 정의한다. 이를 막기 위해, 에이전트가 추론의 종료를 의미하는 토큰을 생성할 확률이 특정 조건을 만족할 때 강제로 추론을 종료시키는 전략을 제안하였고, 이로써 토큰 절약 효과를 보이면서도 정확도를 유지하거나 개선하였다.

3. 인간과의 상호작용 중심의 에이전트 연구

이전 장에서 정리한 것처럼 초기의 도구 에이전트 연구들이 자율성을 높이는 데 많은 초점이 맞춰져 있었다면, 최근에는 인간과의 상호작용을 통해 에이전트의 효용성을 높이는 연구들도 활발히 진행되고 있다. 이러한 상호작용 능력들은 크게 네 가지를 포함한다: 대화 상태 추적 능력, 선제적 정보 수집 능력, 비협조적 사용자 대처 능력, 개인화 능력. 각 능력에 대한 간단한 정의는 표 1에 요약되어 있다. 이어질 서브섹션들에서는 각 능력에 대해 자세히 소개한다.

3.1 대화 상태 추적 능력

사실 인간과 상호작용하며 과업을 수행하는 과업 중심 대화시스템(Task-Oriented Dialogue Systems) 연구는 오늘날의 언어모델이 나오기 한참 전부터 연구되어온 역사 깊은 주제이다 [46–48]. 이 시스템에서 중요한 에이전트의 기능 중 하나는 대화 상태 추적(Dialogue State Tracking) 능력, 즉 언어모델이 대화의 흐름에 따라 사용자로부터 제공되는 요구사항들을 정확히 추적함으로써 자연스러운 대화를 이끌어 가는 것이다 [49, 50]. 대화 상태를 구성하는 중요한 항목으로는 사용자의 의도(예: 식당 예약) 및 관련된 조건들(예: 식당 이름, 예약 시간, 인원수)이 있다. 사용자의 발화로부터 이 정보들을 추출하여 대화 상태를 동적으로 업데이트함으로써 모델은 긴 대화 맥락을 거치면서도 사용자의 필요를 정확히 처리할 수 있다.

그러나 최근 언어모델의 급격한 발전과 함께 언어모델이 특별한 모듈 없이도 대화 맥락을 잘 이해하고 활용할 수 있게 되었고, 이에 따라 대화 상태 추적에 관한 연구는 잠시 주춤하였다. 하지만 도구 에이전트들이 기존의 과업 중심 대화시스템들이 다루던 작은 범위의 서비스에서 크게 확장해 매우 다양하고 이질적인 복잡한 도구들을 처리하게 되면서, 대화 상태 추적을 최신 에이전트에 다시 도입하려는 시도들이 이루어지고 있다 [51].

3.2 선제적 정보 수집 능력

사용자의 지시는 종종 과업 수행에 필요한 정보가 누락되어 있거나 현실적으로 불가능한 내용을 포함하기도 한다. 이러한 상황에서 에이전트가 섣불리 과업을 수행하려고 시도하기보다, 필요한 정보를 판단하고 선제적으로 사용자에게 질문을 던지는 능력은 매우 중요하다. 이는 언어모델이 불확실한 상황에서 성급히 답변하거나 행동하여 발생하는 오류와 환각 현상을 줄이는 데 핵심적인 역할을 한다.

Ask-before-Plan [52] 연구는 에이전트가 계획을 세우기 전에 필요한 정보를 물어보는 정보 수집 능력을 다룬다. 에이전트가 사용자의 지시를 분석하고, 만약 정보가 누락되었거나(예: 여행 계획 요청에 인원수 정보가 없음) API 조회 결과 현실적으로 불가능한 조건(예: 존재하지 않는 날짜의 항공편 예약)이 확인되면, 계획을 세우기 전에 먼저 사용자에게 질문을 통해 문제를 해결한다. 이처럼 사용자와의 대화와 API를 통한 외부 환경과의 상호작용 결과를 종합하여 질문의 필요성을 판단하고, 이를 통해 명확해진 요구사항을 바탕으로 최종 계획을 수립하는 것이 특징이다.

반면, ToolDial [51]은 보다 복잡한 다중 API 호출 과정에서 필요한 정보를 수집하는 능력에 초점을 맞춘다. 예를 들어, 특정 API를 호출하는 데 필요한 정보(예: 항공편 코드)가 있을 때 사용자에게 물어보고, 사용자가 제공하지 못하는 경우 그 정보를 얻기 위해 다른 API를 호출해야 할 수 있다. ToolDial은 이처럼 API 호출에 필요한 정보가 부족할 때, 사용자에게 직접 질문하거나, 사용자가 대답하지 못할 경우 연관된 다른 API를 탐색하여 문제를 해결하는 다단계 상호작용 시나리오를 다룬다. Ask-before-Plan이 주로 사용자의 초기 지시에 담긴 모호성을 해소한다면, ToolDial은 도구 사용 과정에서 발생하는 정보 부족 문제를 사용자와의 대화 및 연쇄적인 API 호출을 통해 해결한다는 점에서 차이가 있다.

이러한 선제적 정보 수집 능력은 사용자와 협력하여 복잡한 결과물을 만들어내는 협동적 추론 작업으로 확장될 수 있다. SWEET-RL [53]은 에이전트가 인간 협력자와 여러 차례 대화를 주고받으며 코드나 웹사이트 디자인 같은 결과물을 함께 만들어 가는 상황을 다룬다. 이 과정에서 에이전트는 최종 결과물을 완성하기 위해 자신에게 어떤 정보가 부족한지(예: 코드의 예외 처리 방식, 디자인의 세부 색상) 스스로 파악하고, 이를 인간 협력자에게 질문을 통해 얻어내야 한다. 이 연구의 핵심은 여러 턴에 걸친 긴 대화 속에서 '어떤 질문이 최종 결과물 완성에 더 도움이 되었는가'를 학습하여, 목표 달성에 기여하는 전략적인 질문을 던지는 능력을 기르는 데 있다. 이는 단발성 질문으로 모호함을 푸는 것을 넘어, 공동의 목표를 달성하기 위한 추론 과정의 일부로서 상호작용을 활용하는 더 높은 수준의 능력을 보여준다.

3.3 비협조적 사용자 대처 능력

대부분의 에이전트 연구들은 협조적인 사용자들을 가정하여 학습되고 평가되어 왔다. 협조적인 사용자란 에이전트가 필요로 하는 정보를 분명하게 전달해주고 과업과 관련 없는 정보는 제공하지 않는 이상적인 사용자를 가리킨다. 하지만 현실 세계에서 사용자들은 에이전트가 제공할 수 없는 서비스를 요청하거나 화를 내는 등 에이전트 입장에서 비협조적인 행동들을 종종 보인다. 따라서 협조적인 사용자를 가정해 개발된 에이전트가 실제

현실에서 어떤 성능을 보일지 알기 어려우며, 실험실 환경에 기반한 고평가가 되어 있을 가능성이 높다.

실제 사용자에 대한 에이전트의 대처 능력을 평가하기 위해 사용자 시뮬레이터가 활용되기 시작되었는데, 초기에는 여전히 협조적인 사용자를 가정했다 [54]. 최근 연구에서는 비협조적 사용자 시뮬레이터 [55]를 제시하는데, 이 시뮬레이터는 도구 에이전트들과 상호작용하며 비협조적 사용자 행동을 보이고 이에 따른 에이전트의 과업 수행 성능을 측정한다. 비협조적 사용자 행동으로는 다음 네 가지를 정의한다: (1) 지원하지 않는 서비스 요청, (2) 과업과 관련 없는 대화 시도, (3) 조급함 표출, (4) 불완전한 발화 제공. 동시에 에이전트가 과업을 수행하는 데 필요한 정보는 철저히 제공되도록 설계되어 있어 이상적인 에이전트라면 과업을 성공할 수 있는 환경을 제공한다. 비협조적 사용자 시뮬레이터를 이용해 다양한 에이전트들의 성능을 측정한 결과, 협조적인 상황에 비해 큰 성능 하락이 관찰되었다. 특히 소형 언어모델을 협조적인 상호작용 데이터셋에 학습한 뒤에 평가하면, 협조적인 상황을 처리하는 능력은 크게 향상하나 비협조적인 상황 대처 능력은 많이 못미치는 것을 확인하였다.

3.4 개인화 능력

개인화 능력은 사용자의 프로필, 선호도, 그리고 현재 처한 환경 등 주어진 맥락을 종합적으로 이해하여 맞춤형으로 도구를 사용하거나 제안하는 능력을 말한다. 이는 단순히 기능적으로 동일한 도구 중 하나를 선택하는 것을 넘어, 누가, 어떤 상황에서 사용하는지를 고려하여 사용자의 만족도를 극대화하는 것을 목표로 한다.

ToolSpectrum [56]은 이러한 개인화된 도구 활용 능력을 평가하기 위한 벤치마크를 제시한다. 동일한 기능을 수행하는 여러 도구(예: 저가형 쇼핑 앱 '테무' 및 빠른 배송의 '쿠팡')가 존재할 때, 에이전트가 사용자의 프로필(예: 학생, 고소득 직장인)과 주변 환경(예: 뇌우, 네트워크 불안정)을 고려하여 가장 적절한 도구를 선택하고 그 이유를 설명하도록 요구한다. 예를 들어, 같은 항공권 예매 요청이라도 사용자가 학생이면 저가 항공을, 날씨가 뇌우이면 기차를 제안하는 식이다. 이 연구는 개인화 요소를 사용자 프로필과 환경이라는 두 가지 축으로 나누어, 각 요소가 도구 선택에 미치는 영향을 체계적으로 분석한다. 실험 결과, 최신 언어모델조차도 두 가지 요소를 동시에 고려하여 추론하는 데 어려움을 겪으며, 이는 개인화가 에이전트 연구에서 아직 해결해야 할 중요한 과제임을 보여준다.

반면, PEToolBench [57]는 명시적 프로필이 아닌 사용자의 상호작용 히스토리를 통해 드러나는 암묵적 선호도 파악 능력을 평가한다. 사용자의 과거 도구 사용 기록을 입력으로 제공하고, 에이전트가 이 히스토리로부터 사용자의 선호도를 추론하여 현재 요청에 적절한 도구를 선택하도록 요구한다. 예를 들어, 사용자가 과거에 주로 쇼핑을 위해 쿠팡을 이용했다면, 사용자의 새로운 구매 요청이 주어졌을 때 쿠팡, 테무 등의 여러 쇼핑 앱 중 쿠팡을 선택해야 한다는 것을 히스토리로부터 유추해야 한다. 에이전트의 도구 선택이 과거 행동 패턴과 일관되는지를 평가함으로써, 명시적 지시 없이 암묵적 선호도를 얼마나 잘 파악하는지 측정한다.

ProPerSim [58]은 개인화와 선제적 제안(Proactivity) 능력을 결합한다. 이 연구는 사용자가 명시적으로 요청하기 전에, 에이전트가 사용자의 상황과 페르소나(성격, 생활 방식 등)를 파악하여 적절한 타이밍에 개인화된 제안을 하는 능력을 평가한다. 가상의 집안 환경에서 생활하는 사용자를 시뮬레이션하고, 인공지능 비서 에이전트가 이 사용자의 행동을 관찰하며 언제, 무엇을 제안할지 결정하게 한다. 예를 들어, 채식주의자인 사용자의 저녁 식사 시간이 가까워지면, 육류가 아닌 채식 메뉴를 선제적으로 추천하는 식이다. 사용자 시뮬레이터는 제안의 내용과 타이밍을 평가하고, 비서 에이전트는 이 피드백을 통해 학습하며 시간이 지남에 따라 더 나은 추천 전략을 발전시킨다.

개인화 능력 측정에 대한 실험 결과들은 최신 언어모델조차도 여러 개인화 요소를 동시에 고려하여 추론하는 데 어려움을 겪는다는 것을 보여준다. 이는 개인화가 에이전트 연구에서 아직 해결해야 할 중요한 과제임을 시사하며, 이러한 연구들은 에이전트가 인간과의 상호작용에서 보다 능동적이고 개인화된 참여자가 될 수 있도록 발전시키는 데 큰 의의가 있다.

4. 논의

본 기고문은 도구 증강 에이전트 연구의 발전 과정을 자율성 중심에서 인간과의 상호작용 중심으로 진화해 온 궤적을 따라 조망했다. 초기 연구들은 에이전트의 자율성을 극대화하기 위해 추론, 계획, 경험 활용, 오류 복구 등 핵심적인 인지 능력들을 강화하는 데 집중했고 아직도 꾸준히 발전하고 있다. 한편 최근에 등장한 연구들은 이러한 자율적 능력을 기반으로 인간과의 상호작용 능력을 고도화하는 방향으로 나아가고 있는데, 특히 사용자의 의도를 이해하고, 먼저 질문하여 정보를 수집하며, 다양한 사용자 유형에 대처하고, 개인의 필요에 맞추는 능력 등에 관한 연구가 활발히 이루어지고 있다. 이는 에이전트가 단독으로 완벽한 결과물을 내놓는 것을 넘어, 인간과 협력하며 더 나은 결과물을 만들어내는 파트너로서의 역할이 중요해지고 있음을 시사한다.

이러한 연구의 흐름을 종합할 때 에이전트의 발전에서 '협력'과 '복잡성'이 점차 강조되고 있음을 볼 수 있다. 도구 에이전트가 인간 수준의 신뢰도와 능력을 갖춘 파트너로 전진하기 위해서는 다음과 같은 연구 과제들이 해결해야 할 숙제로 남아있다.

첫째는 다중 에이전트 협업이다. 이제 에이전트는 인간과의 협업을 넘어 각기 다른 전문성을 가진 에이전트들 간의 협업으로 확장되고 있다. 이처럼 여러 에이전트들이 서로 소통하고, 역할을 분담하며, 갈등을 조정하여 공동의 목표를 달성하게 하는 다

중 에이전트 시스템 연구는 이제 막 시작 단계에 있다. 따라서 에이전트들의 과업 수행 성능을 높이고, 이 과정에서 소통을 효율적으로 만들기 위한 연구들이 수행되어야 한다. 또한 이러한 다중 에이전트 시스템에서 발생할 수 있는 잠재적인 윤리 문제나 법적인 문제에 대해서도 고민이 필요한 시점이다.

둘째는 장기 과업 수행 능력이다. 현재의 에이전트는 비교적 짧고 명확하게 정의된 과업에서는 높은 성공률을 보이지만, 현실 세계처럼 예측 불가능한 변수가 많고 여러 단계에 걸쳐 명확화가 필요한 장기 과업에서는 여전히 취약하다. 가령, 에이전트가 스스로 연구를 진행한다든지, 큰 규모의 서비스나 상품을 설계하고 만드는 등의 작업이 그 예가 될 수 있다. 이를 위해 큰 시야에서 과업을 이해하고, 예상치 못한 오류에 직면했을 때 유연하게 대처하고 적절한 단계로 돌아가며, 변화하는 상황에 맞춰 계획을 동적으로 수정하는 능력은 앞으로의 핵심 연구 주제가 될 것이다.

셋째, 이 외에도 이 기고문에서 언급한 다양한 능력들 중에서 여전히 발전의 여지가 많은 영역들이 있다. 예를 들어, 에이전트의 효율성 강화는 비교적 최근 연구가 시작된 분야로서, 본문에서 소개한 과도한 도구 사용 방지, 도구 실행 시간 단축, 추론을 위한 과도한 토큰 사용 방지 외에도 도구 호출의 안정성 및 API 관련된 여러 가지 제약들을 고려하여 보다 일반화된 형식의 제약된 최적화(Constrained Optimization) 연구로 발전될 가능성이 있다. 또한 사용자 개인화 능력 역시도 단순히 사용자의 취향이나 상황을 고려하는 것에서 더 나아가 사용자의 가치관, 사회적 관계 및 기타 제약들을 고려한 정교한 맞춤화 연구가 진행될 것으로 보인다. 동시에 에이전트의 선제적인 행동을 위해 사용자의 행동이나 주위 상황을 관찰한다든지 해당 기록들을 보관하는 데에서 발생 가능한 잠재적인 문제들이 점차 수면 위로 떠오를 수 있다.

이러한 문제들을 해결하려는 노력은 도구 에이전트를 더욱 신뢰할 수 있고 유능한 파트너로 만들어, 인공지능과 인간의 협업을 새로운 차원으로 이끌 것이다.

AI 사용

본 글은 저자들의 감수 하에 Gemini 2.5 Pro와 ChatGPT 5의 도움(논문 요약)을 받아 작성되었다.

참고 문헌

[1] Google Cloud, "What are AI agents?" 2024, accessed: 2025-10-08. URL: https://cloud.google.com/discover/what-are-ai-agents

[2] IBM, "AI agents," 2024, accessed: 2025-10-08. URL: https://www.ibm.com/think/topics/ai-agents

[3] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in *Advances in Neural Information Processing Systems*, vol. 35. Curran Associates, Inc., 2022, pp. 24 824–24 837. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf

[4] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao, "ReAct: Synergizing Reasoning and Acting in Language Models," in *The Eleventh International Conference on Learning Representations*, 2022. URL: https://openreview.net/forum?id=WE_vluYUL-X

[5] B. Paranjape, S. Lundberg, S. Singh, H. Hajishirzi, L. Zettlemoyer, and M. T. Ribeiro, "ART: Automatic multi-step reasoning and tool-use for large language models," *arXiv*, 2023. URL: https://doi.org/10.48550/arxiv.2303.09014

[6] J. Yang, A. Prabhakar, K. Narasimhan, and S. Yao, "Intercode: Standardizing and benchmarking interactive coding with execution feedback," *Advances in Neural Information Processing Systems*, vol. 36, pp. 23 826–23 854, 2023.

[7] A. Prasad, A. Koller, M. Hartmann, P. Clark, A. Sabharwal, M. Bansal, and T. Khot, "ADaPT: As-Needed Decomposition and Planning with Language Models," *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 4226–4252, 2024. URL: https://doi.org/10.18653/v1/2024.findings-naacl.264

[8] Y. Qin, S. Liang, Y. Ye, K. Zhu, L. Yan, Y. Lu, Y. Lin, X. Cong, X. Tang, B. Qian, S. Zhao, L. Hong, R. Tian, R. Xie, J. Zhou, M. Gerstein, d. li, Z. Liu, and M. Sun, "ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs," in *The Twelfth International Conference on Learning Representations*, 2024. URL: https://openreview.net/forum?id=dHng2O0Jjr

[9] Y. Zhang, H. Cai, X. Song, Y. Chen, R. Sun, and J. Zheng, "Reverse Chain: A Generic-Rule for LLMs to Master Multi-API Planning," *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 302–325, 2024. URL: https://doi.org/10.18653/v1/2024.findings-naacl.22

[10] S. Li, Y. Guo, J. Yao, Z. Liu, and H. Wang, "HomeBench: Evaluating LLMs in Smart Homes with

Valid and Invalid Instructions Across Single and Multiple Devices," *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12 230–12 250, 2025. URL: https://doi.org/10.18653/v1/2025.acl-long.597

[11] E. King, H. Yu, S. Lee, and C. Julien, "Sasha: Creative Goal-Oriented Reasoning in Smart Homes with Large Language Models," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 8, no. 1, pp. 1–38, 2024. URL: https://doi.org/10.1145/3643505

[12] G. Seo, J. Yang, J. Pyo, N. Kim, J. Lee, and Y. Jo, "SimuHome: A Temporal- and Environment-Aware Benchmark for Smart Home LLM Agents," *arXiv*, 2025.

[13] Z. Shi, Y. Wang, L. Yan, P. Ren, S. Wang, D. Yin, and Z. Ren, "Retrieval Models Aren't Tool-Savvy: Benchmarking Tool Retrieval for Large Language Models," *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24 497–24 524, 2025. URL: https://doi.org/10.18653/v1/2025.findings-acl.1258

[14] M. Kachuee, S. Ahuja, V. Kumar, P. Xu, and X. Liu, "Improving Tool Retrieval by Leveraging Large Language Models for Query Generation," in *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*. Abu Dhabi, UAE: Association for Computational Linguistics, 2025, pp. 29–38. URL: https://aclanthology.org/2025.coling-industry.3/

[15] Q. Xu, Y. Li, H. Xia, and W. Li, "Enhancing Tool Retrieval with Iterative Feedback from Large Language Models," in *Findings of the Association for Computational Linguistics: EMNLP 2024*. Miami, Florida, USA: Association for Computational Linguistics, 2024, pp. 9609–9619. URL: https://aclanthology.org/2024.findings-emnlp.561/

[16] S. Hao, T. Liu, Z. Wang, and Z. Hu, "ToolkenGPT: Augmenting Frozen Language Models with Massive Tools via Tool Embeddings," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL: https://openreview.net/forum?id=BHXsb69bSx

[17] R. Wang, X. Han, L. Ji, S. Wang, T. Baldwin, and H. Li, "ToolGen: Unified Tool Retrieval and Calling via Generation," in *The Thirteenth International Conference on Learning Representations*, 2025. URL: https://openreview.net/forum?id=XLMAMmowdY

[18] S. Yuan, K. Song, J. Chen, X. Tan, Y. Shen, K. Ren, D. Li, and D. Yang, "EASYTOOL: Enhancing LLM-based Agents with Concise Tool Instruction," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, L. Chiruzzo, A. Ritter, and L. Wang, Eds. Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 951–972. URL: https://aclanthology.org/2025.naacl-long.44/

[19] S. Lee, N. Kim, and Y. Jo, "In-N-Out: A Parameter-Level API Graph Dataset for Tool Agents," *arXiv*, 2025.

[20] J. Lee, W. Song, J. Han, H. Pyun, and Y. Jo, "Don't Adapt Small Language Models for Tools; Adapt Tool Schemas to the Models," *arXiv*, 2025.

[21] C. Qian, E. C. Acikgoz, Q. He, H. Wang, X. Chen, D. Hakkani-Tür, G. Tur, and H. Ji, "ToolRL: Reward Is All Tool Learning Needs," *arXiv preprint arXiv:2504.13958*, 2025.

[22] K. Chen, M. Cusumano-Towner, B. Huval, A. Petrenko, J. Hamburger, V. Koltun, and P. Krähenbühl, "Reinforcement Learning for Long-Horizon Interactive LLM Agents," *arXiv*, 2025. URL: https://doi.org/10.48550/arxiv.2502.01600

[23] H. Trivedi, T. Khot, M. Hartmann, R. Manku, V. Dong, E. Li, S. Gupta, A. Sabharwal, and N. Balasubramanian, "AppWorld: A Controllable World of Apps and People for Benchmarking Interactive Coding Agents," *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16 022–16 076, 2024. URL: https://doi.org/10.18653/v1/2024.acl-long.850

[24] Y. Song, D. Yin, X. Yue, J. Huang, S. Li, and B. Y. Lin, "Trial and Error: Exploration-Based Trajectory Optimization of LLM Agents," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 7584–7600. URL: https://aclanthology.org/2024.acl-long.409/

[25] P. Putta, E. Mills, N. Garg, S. Motwani, C. Finn, D. Garg, and R. Rafailov, "Agent Q: Advanced Reasoning and Learning for Autonomous AI Agents," 2024. URL: https://arxiv.org/abs/2408.07199

[26] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct Preference Optimization: Your Language Model is Secretly a Reward Model," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL: https://openreview.net/forum?id=HPuSIXJaa9

[27] N. Shinn, F. Cassano, A. Gopinath, K. R. Narasimhan, and S. Yao, "Reflexion: language agents with verbal reinforcement learning," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL: https://openreview.net/forum?id=vAElhFcKW6

[28] A. Zhao, D. Huang, Q. Xu, M. Lin, Y.-J. Liu, and G. Huang, "ExpeL: LLM Agents Are Experiential Learners," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, pp. 19 632–19 642, 2024. URL: https://doi.org/10.1609/aaai.v38i17.29936

[29] C. Qu, S. Dai, X. Wei, H. Cai, S. Wang, D. Yin, J. Xu, and J.-R. Wen, "From Exploration to Mastery: Enabling LLMs to Master Tools via Self-Driven Interactions," in *The Thirteenth International Conference on Learning Representations*, ser. arXiv, 2024. URL: https://openreview.net/forum?id=QKBu1BOAwd

[30] A. Kumar, V. Zhuang, R. Agarwal, Y. Su, J. D. Co-Reyes, A. Singh, K. Baumli, S. Iqbal, C. Bishop, R. Roelofs, L. M. Zhang, K. McKinney, D. Shrivastava, C. Paduraru, G. Tucker, D. Precup, F. Behbahani, and A. Faust, "Training Language Models to Self-Correct via Reinforcement Learning," 2024. URL: https://arxiv.org/abs/2409.12917

[31] H. Wang, C. T. Leong, J. Wang, and W. Li, "E^2CL: Exploration-based Error Correction Learning for Embodied Agents," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 7626–7639. URL: https://aclanthology.org/2024.findings-emnlp.448/

[32] S. Yuan, Z. Chen, Z. Xi, J. Ye, Z. Du, and J. Chen, "Agent-R: Training Language Model Agents to Reflect via Iterative Self-Training," *arXiv*, 2025. URL: https://doi.org/10.48550/arxiv.2501.11425

[33] Y. Ruan, H. Dong, A. Wang, S. Pitis, Y. Zhou, J. Ba, Y. Dubois, C. J. Maddison, and T. Hashimoto, "Identifying the Risks of LM Agents with an LM-Emulated Sandbox," in *The Twelfth International Conference on Learning Representations*, 2024. URL: https://openreview.net/forum?id=GEcwtMk1uA

[34] Q. Zhan, Z. Liang, Z. Ying, and D. Kang, "INJECAGENT: Benchmarking Indirect Prompt Injections in Tool-Integrated Large Language Model Agents," in *Findings of the Association for Computational Linguistics: ACL 2024*. Bangkok, Thailand: Association for Computational Linguistics, 2024. URL: https://aclanthology.org/2024.findings-acl.624/

[35] E. Debenedetti, J. Zhang, M. Balunovic, L. Beurer-Kellner, M. Fischer, and F. Tramèr, "AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents," in *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, Eds., vol. 37. Curran Associates, Inc., 2024, pp. 82 895–82 920. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/97091a5177d8dc64b1da8bf3e1f6fb54-Paper-Datasets_and_Benchmarks_Track.pdf

[36] M. Andriushchenko, A. Souly, M. Dziemian, D. Duenas, M. Lin, J. Wang, D. Hendrycks, A. Zou, Z. Kolter, M. Fredrikson, E. Winsor, J. Wynne, Y. Gal, and X. Davies, "AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents," *arXiv preprint arXiv:2410.09024*, 2024, preprint. URL: https://arxiv.org/abs/2410.09024

[37] H. Zhang, J. Huang, K. Mei, Y. Yao, Z. Wang, C. Zhan, H. Wang, and Y. Zhang, "Agent Security Bench (ASB): Formalizing and Benchmarking Attacks and Defenses in LLM-based Agents," in *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025, iCLR 2025. URL: https://openreview.net/forum?id=V4y0CpX4hK

[38] Z. Xiang, L. Zheng, Y. Li, J. Hong, Q. Li, H. Xie, J. Zhang, Z. Xiong, C. Xie, C. Yang, D. Song, and B. Li, "GuardAgent: Safeguard LLM Agents via Knowledge-Enabled Reasoning," in *Forty-second International Conference on Machine Learning*, 2025. URL: https://openreview.net/forum?id=2nBcjCZrrP

[39] W. Hua, X. Yang, M. Jin, Z. Li, W. Cheng, R. Tang, and Y. Zhang, "TrustAgent: Towards Safe and Trustworthy LLM-based Agents," in *Findings of the Association for Computational Linguistics: EMNLP 2024*. Miami, USA: Association for Computational Linguistics, 2024. URL: https://aclanthology.org/2024.findings-emnlp.585/

[40] Y. Huang, J. Shi, Y. Li, C. Fan, S. Wu, Q. Zhang, Y. Liu, P. Zhou, Y. Wan, N. Z. Gong, and L. Sun, "MetaTool Benchmark for Large Language Models: Deciding Whether to Use Tools and Which to Use," in *The Twelfth International Conference on Learning Representations*, 2024. URL: https://openreview.net/forum?id=R0c2qtalgG

[41] W. Li, D. Li, K. Dong, C. Zhang, H. Zhang, W. Liu, Y. Wang, R. Tang, and Y. Liu, "Adaptive Tool Use in Large Language Models with Meta-Cognition Trigger," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 13 346–13 370. URL: https://aclanthology.org/2025.acl-long.655/

[42] C. Qian, E. C. Acikgoz, H. Wang, X. Chen, A. Sil, D. Hakkani-Tür, G. Tur, and H. Ji, "SMART: Self-Aware Agent for Tool Overuse Mitigation," in *Findings of the Association for Computational Linguistics: ACL 2025*, 2025, pp. 4604–4621. URL: https://aclanthology.org/2025.findings-acl.239/

[43] D. Wu, J. Wang, Y. Meng, Y. Zhang, L. Sun, and Z. Wang, "CATP-LLM: Empowering Large Language Models for Cost-Aware Tool Planning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[44] S. Kim, S. Moon, R. Tabrizi, N. Lee, M. W. Mahoney, K. Keutzer, and A. Gholami, "An LLM Compiler for Parallel Function Calling," in *Forty-first International Conference on Machine Learning*, 2024. URL: https://openreview.net/forum?id=uQ2FUoFjnF

[45] M. Oh, S. Song, S. Lee, S. Jo, and Y. Jo, "ThinkBrake: Mitigating Overthinking in Tool Reasoning," *NeurIPS 2025 Efficient Reasoning Workshop*, 2025. URL: https://arxiv.org/abs/2510.00546

[46] D. G. Bobrow, R. M. Kaplan, and M. Kay, "GUS: A Frame-Driven Dialog System," *Artificial Intelligence*, vol. 8, no. 2, pp. 155–173, 1977. URL: https://doi.org/10.1016/0004-3702(77)90018-2

[47] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The ATIS Spoken Language Systems Pilot Corpus," in *Proceedings of the Workshop on Speech and Natural Language*, 1990, pp. 96–101. URL: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/atis.pdf

[48] L. E. Asri, H. Schulz, S. Sharma, J. Zumer, J. Harris, E. Fine, R. Mehrotra, and K. Suleman, "Frames: a corpus for adding memory to goal-oriented dialogue systems," in *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, K. Jokinen, M. Stede, D. DeVault, and A. Louis, Eds. Saarbrücken, Germany: Association for Computational Linguistics, Aug. 2017, pp. 207–219. URL: https://aclanthology.org/W17-5526/

[49] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gašić, "MultiWOZ – A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, pp. 5016–5026. URL: https://aclanthology.org/D18-1547/

[50] A. Rastogi, X. Zang, S. Sunkara, R. Gupta, and P. Khaitan, "Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 8689–8696. URL: https://arxiv.org/abs/1909.05855

[51] J. Shim, G. Seo, C. Lim, and Y. Jo, "ToolDial: Multi-turn Dialogue Generation Method for Tool-Augmented Language Models," in *The Thirteenth International Conference on Learning Representations*, 2025. URL: https://openreview.net/forum?id=J1J5eGJsKZ

[52] X. Zhang, Y. Deng, Z. Ren, S.-K. Ng, and T.-S. Chua, "Ask-before-Plan: Proactive Language Agents for Real-World Planning," *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10 836–10 863, 2024. URL: https://doi.org/10.18653/v1/2024.findings-emnlp.636

[53] Y. Zhou, S. Jiang, Y. Tian, J. Weston, S. Levine, S. Sukhbaatar, and X. Li, "SWEET-RL: Training Multi-Turn LLM Agents on Collaborative Reasoning Tasks," *arXiv*, 2025. URL: https://doi.org/10.48550/arxiv.2503.15478

[54] S. Yao, N. Shinn, P. Razavi, and K. R. Narasimhan, "τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains," in *The Thirteenth International Conference on Learning Representations*, 2025. URL: https://openreview.net/forum?id=roNSXZpUDN

[55] J. Shim, W. Song, C. Jin, S. KooK, and Y. Jo, "Non-Collaborative User Simulators for Tool Agents," *arXiv*, 2025. URL: https://doi.org/10.48550/arxiv.2509.23124

[56] Z. Cheng, H. Wang, Z. Liu, Y. Guo, Y. Guo, Y. Wang, and H. Wang, "ToolSpectrum: Towards Personalized Tool Utilization for Large Language Models," *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20 679–20 699, 2025. URL: https://doi.org/10.18653/v1/2025.findings-acl.1063

[57] Q. Xu, Y. Li, H. Xia, F. Liu, M. Yang, and W. Li, "PEToolLLM: Towards Personalized Tool Learning in Large Language Models," in *Findings of the Association for Computational Linguistics: ACL 2025*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 21 488–21 503. URL: https://aclanthology.org/2025.findings-acl.1107/

[58] J. Kim, J. Choi, W. Chay, D. Kyung, Y. Kwon, Y. Jo, and E. Choi, "ProPerSim: Developing Proactive and Personalized AI Assistants through User-Assistant Simulation," *arXiv*, 2025. URL: https://doi.org/10.48550/arxiv.2509.21730