대규모 언어 모델 기반 스마트폰 앱 사용 자동화 기술 동향

Trends in Technology for Automating the Use of Smartphone Apps Based on Large Language Models

최수길 (S.G. Choi, sooguri@etri.re.kr) 감각확장연구실 책임연구원
정치윤 (C.Y. Jeong, iamready@etri.re.kr) 감각확장연구실 책임연구원/실장
김무섭 (M.S. Kim, gomskim@etri.re.kr) 감각확장연구실 책임연구원/기술총괄

ABSTRACT

Smartphone usage can improve life satisfaction in old age; however, older adults still face difficulties in accessing the services they need in everyday life through their smartphones. Frustration caused by difficult-to-understand icons or menus, anxiety about problems that may arise from incorrect choices, and the desire to avoid losing time due to unfamiliar app usage are obstacles that prevent older adults from using apps that offer useful services. App usage automation technology has gained attention as a way to assist older adults with smartphone app usage. Recently, research has been actively conducted on app usage automation agents that use large language models (LLMs) capable of performing high-level reasoning and planning. In this paper, we introduce the latest research trends in LLM-based app usage automation technology and discuss the necessary research directions for effectively applying these technologies as practical app usage assistance tools.

KEYWORDS LLM, 대규모 언어 모델, 앱 UI 이해, 앱 사용 보조, 앱 사용 자동화

Ⅰ. 서론

'2023년 노인실태조사'에 따르면 65세 이상 노인의 스마트폰 보유 비율은 76.6%이고, 하루 평균 1.3시간 스마트폰을 사용하고 있다[1]. 노인의 디지털 정보 활용 능력이 삶의 만족도에 긍정적인 영향을 미치며, 자기효능감을 매개하여 노년기 삶의 만족도를 높인다는 연구 결과도 있다[2]. 하지만, 고령자는 스마트폰을 이용해서 실생활에 필요한 서비스를 받는데 여전히 어려움을 겪고 있다. '2023년 서울시민 디지털역량실태조사'에 따르면 고령층의 메신저(87.9%), 정보검색(86.9%), 동영상 시청(81.0%) 이용

* DOI: https://doi.org/10.22648/ETRI.2025.J.400207
* 본 연구는 한국전자통신연구원 연구운영비지원사업(기본사업)의 일환으로 수행되었음[25ZB1200, 인간중심의 자율지능시스템 원천기술연구].

경험률은 전체 평균 대비 큰 차이가 없는 반면, 상품구매(38.4%), 음식배달(30.0%), 교통/서비스예약(27.4%) 등 실생활 밀착 분야의 비대면 서비스 경험률은 상대적으로 낮게 나타났다[3].

위와 같이 고령자의 사용률이 낮은 서비스를 제공하는 스마트폰 앱은 상대적으로 UI(User Interface)가 복잡하고, 선택하거나 입력해야 하는 정보가 많은 특징이 있다. 이해하기 어려운 아이콘이나 메뉴로 인한 답답함, 잘못된 선택으로 발생하는 문제에 대한 불안감, 그리고 익숙하지 않은 앱 사용에 걸리는 시간 손실을 피하고 싶은 마음이 유용한 서비스를 제공하는 앱의 사용을 막는 걸림돌이 되고 있다.

고령층의 스마트폰 사용을 돕기 위해서 노인복지관 등의 공공기관에서 스마트폰 사용 교육 프로그램을 제공하고 있으며, 통신 업계도 스마트폰 활용 방법 안내 서비스를 통해서 고령층 고객을 공략하고 있다. 하지만, 고령자 대상 스마트폰 교육 담당자와의 인터뷰에 의하면 교육을 받고자 하는 수요에 비해서 개설된 교육 과정이 부족하며, 교육 후 지속적으로 스마트폰을 사용하지 않으면 교육 효과가 급격히 사라지는 문제가 있다.

따라서, 교육뿐만 아니라 스마트폰 사용에서 어려움에 부딪혔을 때 즉각적으로 도움을 줄 수 있는 환경의 구축이 필요하다. 빠르게 도움을 줄 수 있는 인적 서비스가 있으면 좋지만 비용 측면에서 한계가 명확하고, 인공지능 챗봇을 이용한 도움 서비스가 늘어나고 있지만 챗봇과 대화를 통해서 필요한 도움을 끌어내는 것도 도전과제이다. 표 1[4]과 같이 모바일 웹 · 앱을 설계할 때 준수해야 하는 지침이 있지만, 강제성이 없고 복잡한 서비스에 적용하기 어려운 문제가 있다.

표 1 고령층 친화 디지털 접근성 표준 10대 지침

1. 글자는 크고 선명해야 합니다.
2. 필수적인 요소로 구성되어야 합니다.
3. 정보구조는 단순하고 친숙해야 합니다.
4. 용어는 이해하기 쉬워야 합니다.
5. 시스템 상태는 가시적이어야 합니다.
6. 조작기능(컨트롤)은 행동을 유발해야 합니다.
7. 조작기능(컨트롤)은 신속하고 정확하게 조작할수 있어야 합니다.
8. 조작결과(피드백)를 제공해야 합니다.
9. 사용자 오류를 예방하고 복구할 수 있어야 합니다.
10.심리적 부담을 줄여야 합니다.

출처 Reprinted from 서울디지털재단, "고령층 친화 디지털 접근성 표준," 2021. https://sdf.seoul.kr/research-report/1458

스마트폰 앱 사용을 보조하기 위한 다른 방안으로 앱 사용을 자동화하는 기술이 연구되고 있다. 앱 사용 자동화는 필요한 메뉴의 선택을 스스로 판단하고 정보 입력에 필요한 사용자 개입을 최소화해서 사용자가 필요한 서비스를 누릴 수 있도록 보조하는 에이전트의 사용을 의미한다. 이러한 자동화 에이전트의 구현을 위해서 사용자 정의 Task를 사전 정의된 API(Application Programming Interface)의 조합으로 구성하는 방식은 확장성이 낮고 복잡한 Task를 표현하기에는 어려움이 있다.

최근에는 고수준의 추론과 계획 등을 수행할 수 있는 대규모 언어 모델(LLM: Large Language Model)을 이용한 앱 사용 자동화 에이전트의 연구가 활발히 이루어지고 있다. 이에 본고에서는 LLM 기반 앱 사용 자동화 기술의 최신 연구 동향을 소개하고, 실제 앱 사용 보조 도구로 본격적으로 활용하기 위해서 필요한 연구 방향에 대하여 논의하고자 한다.

II. LLM 기반 앱 사용 자동화 기술

앱 사용을 자동화하기 위해서는 사용자의 앱 사용 의도 이해 지능, 앱 UI 이해 지능, 그리고 앱 UI 실행 계획 지능이 필요하다. LLM을 이용하는 기본적인 방법은 앱 UI를 텍스트로 표현한 것과 앱 사용 의도를 텍스트로 표현한 것을 같이 입력으로 받고, UI의 어떤 요소를 선택하거나 어떤 정보를 입력할지와 같은 UI 실행 계획을 출력하는 LLM을 학습하

는 것이다. 이러한 연구의 흐름에서 널리 이용되는 데이터셋을 공개하고 베이스라인 시험 결과를 제시한 연구를 1절에서 소개하고, 자동화 정확도를 높이기 위한 개선 연구를 2절에서 정리한다.

1. Android in the Wild

LLM 기반 앱 사용 자동화 에이전트의 학습을 위해서는 대량의 데이터가 필요하고, 참고문헌[5]에서 공개한 AITW(Android in the Wild) 데이터셋이 널리 이용된다. AITW 데이터셋은 GOOGLEAPPS, WEBSHOPPING, INSTALL, GENERAL, SINGLE의 5개 세부 카테고리로 구성된다. 데이터셋의 구성 설명에 일반적으로 사용하는 용어는 표 2와 같다.

AITW 데이터셋은 30,378개 Instruction과 715,142개 Episode로 구성된다. 웹 검색과 같이 매번 결과가 다를 수 있는 Task는 Instruction이 같아도 Episode를 구성하는 Screen과 Action은 다를 수 있어서 한 개의 Instruction에 복수의 Episode를 생성할 수 있다. Task를 구성하는 Step(UI 변화 또는 화면의 전환이 생기는 경우)이 복수이면 Multi-Step Task이며, 그렇지 않으면 Single-Step Task로 분류한다. GOOGLEAPPS, WEBSHOPPING, INSTALL, GENERAL 카테고리는 Multi-Step Task에 해당한다.

그림 1은 AITW 데이터셋의 INSTALL 카테고리에서 Uninstall “Google News” Instruction에 해당하

표 2 앱 사용 자동화 데이터셋 구성 요소 설명

용어	설명
Instruction	사용자가 받고자 하는 서비스(앱 사용 목적)를 표현한 텍스트
Screen	버튼이나 메뉴 등을 선택하면 변하거나 새로 나타나는 UI 화면
Action	버튼 또는 텍스트 입력 등의 Action Type, Screen에서 좌표, 텍스트 입력의 경우에 입력할 텍스트로 구성
Episode	Task 완료에 필요한 전체 Step

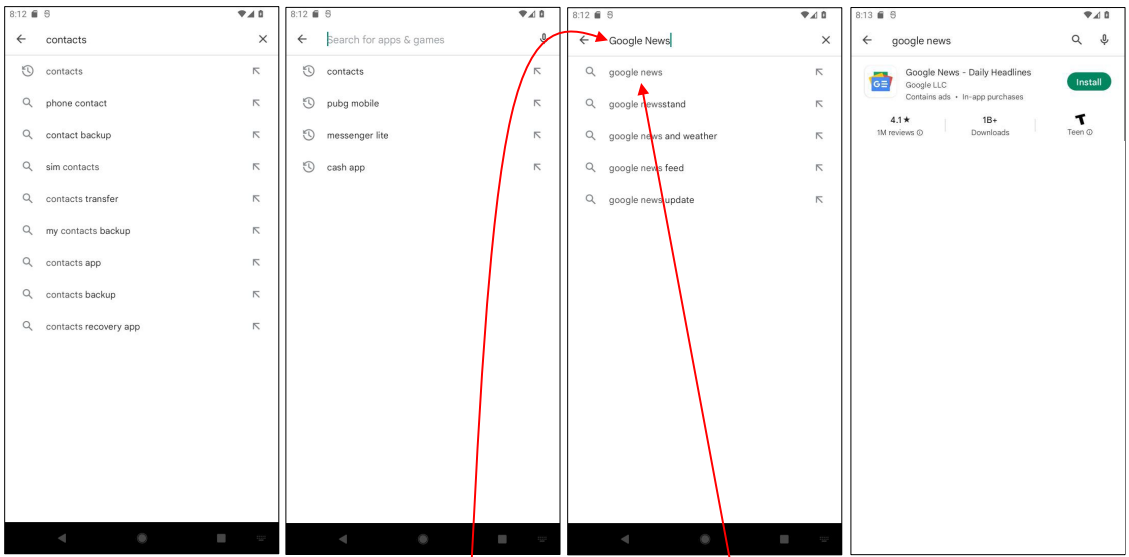


그림 1 AITW 데이터셋 INSTALL 카테고리에서 uninstall “Google News” Instruction에 해당하는 Episode 구성과 Action Prediction 예

는 Episode의 Step3부터 Step6까지의 Screen Shot과 Predicted Action을 보여준다. "Google News" 앱을 찾기 위한 입력이 자동으로 실행되고, 검색 결과로 나온 "Google News"를 선택해서 설치 관련 화면이 나타난다.

참고문헌[5]에서 데이터셋 평가를 위해서 사용한 자동화 에이전트는 그림 2와 같이 동작한다. PaLM 2[6] LLM을 사용하며, 현재 화면의 UI는 HTML syntax를 이용하여 표현된다. AITW 데이터셋에 UI 구성 정보가 포함되어 있기 때문에 UI 분석 과정은 실제로 일어나지는 않는다.

자동화 에이전트의 성능 평가 Metric으로 Action Accuracy와 Task Success Rate가 주로 사용된다. Multi-Step Task가 실패인 경우에도 Task를 구성하는 일부 Action은 성공일 수 있기 때문에 Action Accuracy가 Task Accuracy보다 높은 경향을 보인다. 성공 여부를 판단함에 있어서 자동화 에이전트의 Action Prediction이 Ground Truth Action과 정확히 일치하기 어렵기 때문에 일정 수준의 오차는 허용하는 metric을 이용하여 학습하고 테스트한다.

표 3은 참고문헌[5]에서 제안한 자동화 에이전트의 Action Accuracy를 나타낸다. BC(Behavioural Cloning)는 참고문헌[5]에서 제안한 Transformer 기반의 에이전트이며 LLM 기반 에이전트보다 더 좋은 성능을 보인다. 표 3에서 BC-history는 현재 Episode에서 직전 Step까지 UI와 Action 정보를 추가로 사용한 결과이다.

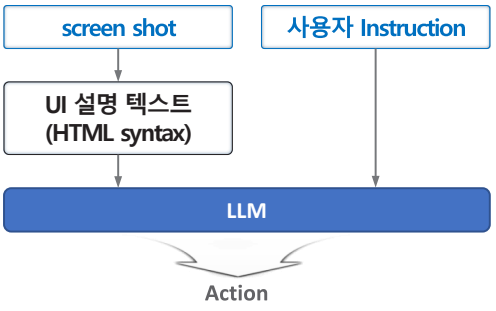


그림 2 LLM 기반 자동화 에이전트 동작 흐름

표 3 AITW 데이터셋에 대한 성능 시험 결과

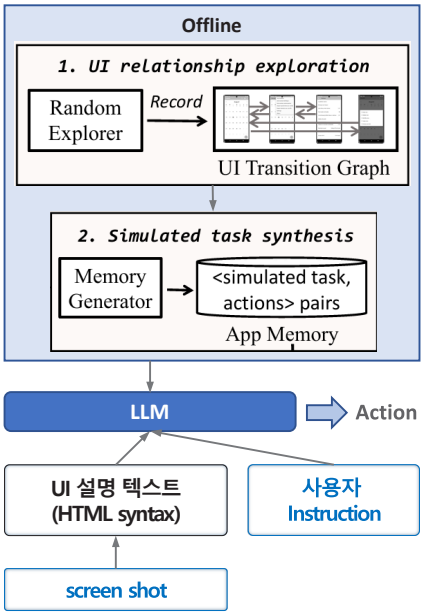
Action accuracy		
BC-single	BC-history	LLM
68.7	73.1	39.6

출처 Reproduced from C. Rawles et al., "Android in the Wild: A Large-Scale Dataset for Android Device Control," arXiv preprint, 2023. doi: 10.48550/arXiv.2307.10088

2. AutoDroid

일반적으로 LLM은 특정 분야에 특화된 데이터가 아니고 다양한 분야의 데이터로 학습을 하고, AITW 데이터셋과 같이 앱 사용에 특화된 데이터로 LLM을 Fine-Tuning 해서 앱 사용 자동화 에이전트가 만들어진다. 이렇게 생성된 자동화 에이전트는 상식적 지식(Commonsense Knowledge)과 Fine-Tuning에 사용된 앱에 대한 지식을 갖고 있지만 학습되지 않은 앱에 적용 시 성능 저하가 나타날 수 있다.

이러한 문제를 해결하기 위하여 AutoDroid[7]는 그림 3과 같이 특정 앱에 대한 사전 탐색을 통하여 UTG(UI Transition Graph)를 생성하고, 현재 진행 중인 Task와 유사한 정보를 UTG에서 추출하여 LLM에 입력함으로써 Action Prediction 정확도를 높인다. 참고문헌[7]의 시험 결과에 따르면 사전 탐색 정보를 GPT-4에 추가로 입력 시 Action Accuracy가 2.7% 상승하고, 소형 Language Model인 Vicuna 7B[8]에 입력 시 53.4% 상승한다. 이 시험 결과에서 주목할 점은 GPT-4와 같이 일반 추론 성능이 높은 LLM을 사용하면 특정 앱에 대한 사전 지식이 없어도 높은 정확도를 보이지만, 소형 Language Model에 기반한 자동화 에이전트의 정확도를 높이기 위해서는 사전 탐색 정보의 활용이 중요한 역할을 할 수 있다는 것이다.



출처 Reproduced from H. Wen et al., "AutoDroid: LLM-powered Task Automation in Android," in Proc. Int. Conf. Mobile Comput. Netw., (Washington D.C. DC USA), Nov. 2024, pp. 543-557.

그림 3 앱 사전 탐색을 포함하는 AutoDroid 동작 흐름

AutoDroid[7]는 자체 제작 데이터셋에 대한 시험 결과만 제공하므로 LLM 기반의 다른 방법과 직접적인 성능 비교는 어렵다. AutoDroid[7]는 사전 탐색을 통하여 특정 앱에 대한 지식 생성 방법 외에 소형 Language Model의 성능 향상을 위한 Fine-Tuning과 처리 속도 향상을 위한 Query Optimization 방법도 제안한다.

Ⅲ. VLM 기반 앱 사용 자동화 기술

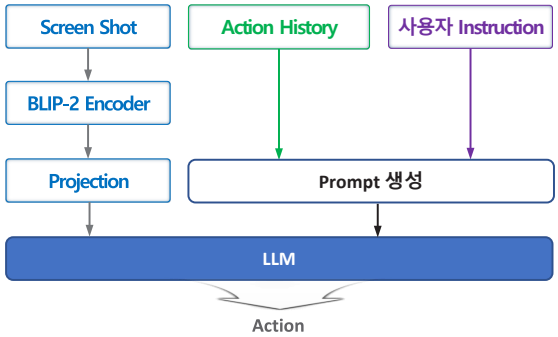
앱 UI 정보를 이용하는 방법은 두 가지로 구분할 수 있다. 아이콘 탐지와 OCR(Optical Character Recognition) 기술 등을 이용해서 앱 화면의 구성 정보를 파악하고 텍스트로 표현하여 자동화 에이전트에 입력하는 것이 첫 번째 방법이고, Ⅱ장의 LLM 기반 자동화 에이전트에 사용된다. 앱 화면을 구성하는 세부 정보를 파악하는 구체적인 방법과 도구는 다양하며, VisionTasker[9], Mobile-Agent[10] 그리고 Mobile-Agent-v2[11]를 참고할 수 있다.

두 번째는 앱 Screen shot 이미지를 VLM(Vision Language Model)의 입력으로 사용하는 것이다. UI 정보 획득을 위한 별도의 단계를 거치지 않기 때문에 상대적으로 구현이 단순하고, 아이콘 탐지와 OCR 등에서 발생할 수 있는 오류의 영향을 받지 않는다는 장점이 있다. 최근 주요 학회에 발표된 VLM 기반 앱 사용 자동화 에이전트는 Auto-UI[12], Cog Agent[13], CoCo-Agent[14]가 대표적이다.

1. Auto-UI

Auto-UI[12]는 그림 4와 같이 앱 UI를 텍스트로 표현하지 않고 앱 Screen Shot 이미지를 BLIP-2[15]로 Encoding 후에 Projection Layer를 거쳐서 LLM에 입력한다. LLM은 Encoder-Decoder 구조의 T5[16]를 Alpaca 데이터셋으로 Fine-Tuning한 것을 사용한다.

Action history는 각 Episode에서 직전 Step까지 발생한 Action 정보를 의미한다. AITW[5]의 BC 에이전트도 Action History를 사용하기 때문에 Action History의 사용이 VLM 기반 자동화 에이전트의 고



출처 Reproduced from Z. Zhang et al., "You Only Look at Screens: Multimodal Chain-of-Action Agents," in Proc. Assoc. Comp. Linguistics., (Bangkok, Thailand), Aug. 2024, pp. 3132-3149.

그림 4 Auto-UI 자동화 에이전트 동작 흐름

유한 특징은 아니다. Auto-UI[12]에서 사용하는 Action History는 Action Type(클릭, 텍스트 입력 등), 선택된 UI element의 좌표, Action Type이 텍스트 입력인 경우에 입력된 텍스트로 구성된다. 직전 몇 개 Step의 Action 정보를 사용할지도 성능에 영향을 주게 되며, 직전 8개 Step까지 Action 정보를 사용하면 AITW 데이터셋에 대한 Action Accuracy가 약 5% 향상하는 것으로 나타났다. Action history를 어떻게 표현하는지도 성능에 영향을 줄 수 있으며, Vision-Tasker[9]와 CoCo-Agent[14]는 선택된 UI element의 좌표 뿐만 아니라 선택된 Element의 설명까지 Action History로 사용하는 차이가 있다.

단, Auto-UI[12]의 공개 코드를 분석해 보면 Ground-Truth Action을 History로 사용하는 문제가 발견된다. 실생활에서 앱의 사용을 자동화하기 위해서는 Predicted Action을 History로 사용해야 하고, Prediction에 오류가 있을 수 있기 때문에 Auto-UI[12] 논문의 실험 결과와 같은 정확도 향상이 어려울 수도 있다. Predicted Action을 History로 사용하면 정확도가 향상되는지 여부는 시험을 통해 확인이 필요하다.

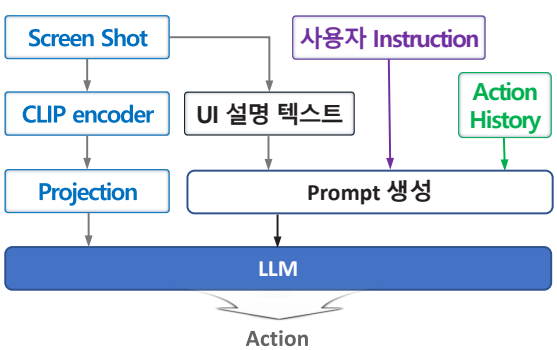
2. CogAgent

CogAgent[13]는 Auto-UI[12]와 비교하여 Action History를 사용하지 않고, Screen Shot 이미지 Encoding 과정을 고해상도와 저해상도 이미지 Encoding으로 세분화하는 차이가 있다. 작은 아이콘과 텍스트 정보를 활용하기 위해서는 고해상도 이미지를 사용해야 하지만 이미지Token Size를 증가시켜서 처리 속도 저하의 원인이 된다. 가령, 1,120×1,120 이미지를 14×14 patch size로 처리하면 6,400개의 Token이 생성되고, 224×224 이미지는 256개의 Token 이 생성된다. Transformer 기반의 Language Model은 입력 Token Size의 제곱에 비례하여 처리 시간이 증가하는 경향이 있으므로 이미지 해상도가 5배 증가하면 처리 시간은 625배 증가하는 것으로 단순 계산할 수 있다.

고해상도 이미지를 사용하면서 처리 속도 저하를 줄이기 위해서 약 0.3B parameter의 이미지 Encoder를 이용한다. 저해상도 이미지 Encoder는 4.4B Parameter를 가지는 것과 비교하여 매우 경량형 Encoder를 사용함을 알 수 있다. 경량 이미지 Encoder 사용으로 인한 성능 한계를 극복하기 위하여 고해상도 이미지 Encoding 결과를 저해상도 이미지를 입력으로 받는 VLM과 Cross-Attention을 이용하여 융합하는 방법을 제안한다.

3. CoCo-Agent

CoCo-Agent[14]는 Auto-UI[12]와 비교하여 BLIP-2[15] 이미지 Encoder를 CLIP[17]으로 변경하고, UI를 설명하는 텍스트를 추가로 LLM에 입력하는 차이가 있다. CoCo-Agent[14]의 동작 흐름은 그림 5와 같다. 그리고 AITW[5] 데이터셋에서는 Touch Point와 Lift Point가 같은 Tab Action도 두 개



출처 Reproduced from X. Ma et al., "CoCo-Agent: A Comprehensive Cognitive MLLM Agent for Smartphone GUI Automation," in Proc. Assoc. Comp. Linguistics., (Bangkok, Thailand), Aug. 2024, pp. 9097-9110.

그림 5 CoCo-Agent 동작 흐름

표 4 VLM 기반 앱 사용 자동화 에이전트의 AITW 데이터셋에 대한 성능 시험 결과

자동화 에이전트	Action accuracy	Language Model
Auto-UI[12]	74.27	flan-alpaca-0.7B
CogAgent[13]	76.88	Vicuna-7B
CoCo-Agent[14]	79.05	Llama-2 chat-7B

Point를 Prediction하도록 하지만, CoCo-Agent[14]는 한 개 Point만 Prediction해서 오류 발생 소지를 줄인다.

4. 성능 비교 분석

VLM 기반 앱 사용 자동화 에이전트의 AITW[5] 데이터셋에 대한 시험에서 표 4와 같이 CoCo-Agent[14]가 가장 높은 Action Accuracy를 보인다. CoCo-Agent[14]는 앱 Screen Shot 이미지와 UI 텍스트 표현 정보를 모두 이용하는 것이 성능 향상에 주효했을 수 있다. 하지만, 사용된 Language Model이 다르므로 에이전트별로 특별한 세부 방법이 성능에 미치는 영향을 확인하기 위해서는 동일한 Language Model을 사용하는 시험이 필요하다. 그리고, Cog Agent[13]는 AITW[5] 데이터셋으로 학습하기 전에 자체 제작한 데이터셋으로 Pre-Training을 한다. 따라서 Pre-Training 단계 없이 학습한 모델로 성능 비교를 해야 에이전트별로 강점으로 제시하는 방법의 상대적 효용성을 확인할 수 있다.

Ⅳ. 현 수준 분석과 발전 방향

1. 앱 사용 자동화 수준

지금까지 살펴본 자동화 에이전트들의 성능 평가를 위해서 사용하는 AITW[5] 데이터셋의 WEBSHOPPING과 GENERAL 카테고리에서는 정확도가 상대적으로 낮게 나타난다. CoCo-Agent[14]의 WEBSHOPPING 카테고리에 대한 Action Accuracy는 75%이다. WEBSHOPPING 카테고리의 Task는 그림 1과 같이 검색 후 선택으로 이어지는 흐름이 유사하고, 검색어 입력창도 화면 상단의 유사한 위치에 있는 경우가 많다.

하지만, 참고문헌[3]의 통계 자료에 나타난 것처럼 고령인이 사용에 어려움을 겪는 교통/서비스 예약의 대표적인 앱인 코레일톡을 살펴보면 복잡도의 차이가 큼을 알 수 있다. 그림 6은 코레일톡 앱을 이용한 기차표 예매에서 일부 화면을 보여준다. AITW[5] 데이터셋에 대한 시험에서는 고려하지 않아도 되는 여러 가지 어려움이 파악되며, 대표적인 몇 가지를 정리하면 다음과 같다.

- 필요한 정보가 한 화면에 나타나지 않아서 화면 또는 특정 영역을 Swipe하고, 필요한 정보가 화면에 보이는 시점을 인식할 수 있어야 함
- 결제나 로그인 등을 위해서 복수의 정보를 해당 항목에 정확히 입력할 수 있어야 함. 입력할 정보가 사용자 Instruction에 명시적으로 포함되어 있지 않으면 자동 입력이 어려울 수 있음
- 사용자가 요청한 Task를 완료했는지 판단할 수 있어야 함. AITW[5] 데이터셋은 Episode의 끝까지 진행하면 완료이므로 Task 완료 여부를 판단하지 않음

자동화 에이전트의 성능 평가에 사용되는 앱과 실제 활용성이 있는 앱 간의 괴리가 크기 때문에 현실성 있는 데이터셋을 구축하고 학습 및 시험 평가를 하는 것이 중요하다.

2. 스마트폰에 자동화 에이전트 구현

본고에서 설명한 자동화 에이전트들은 사용자 시험을 위해서 ADB(Android Debug Bridge)를 이용한다.

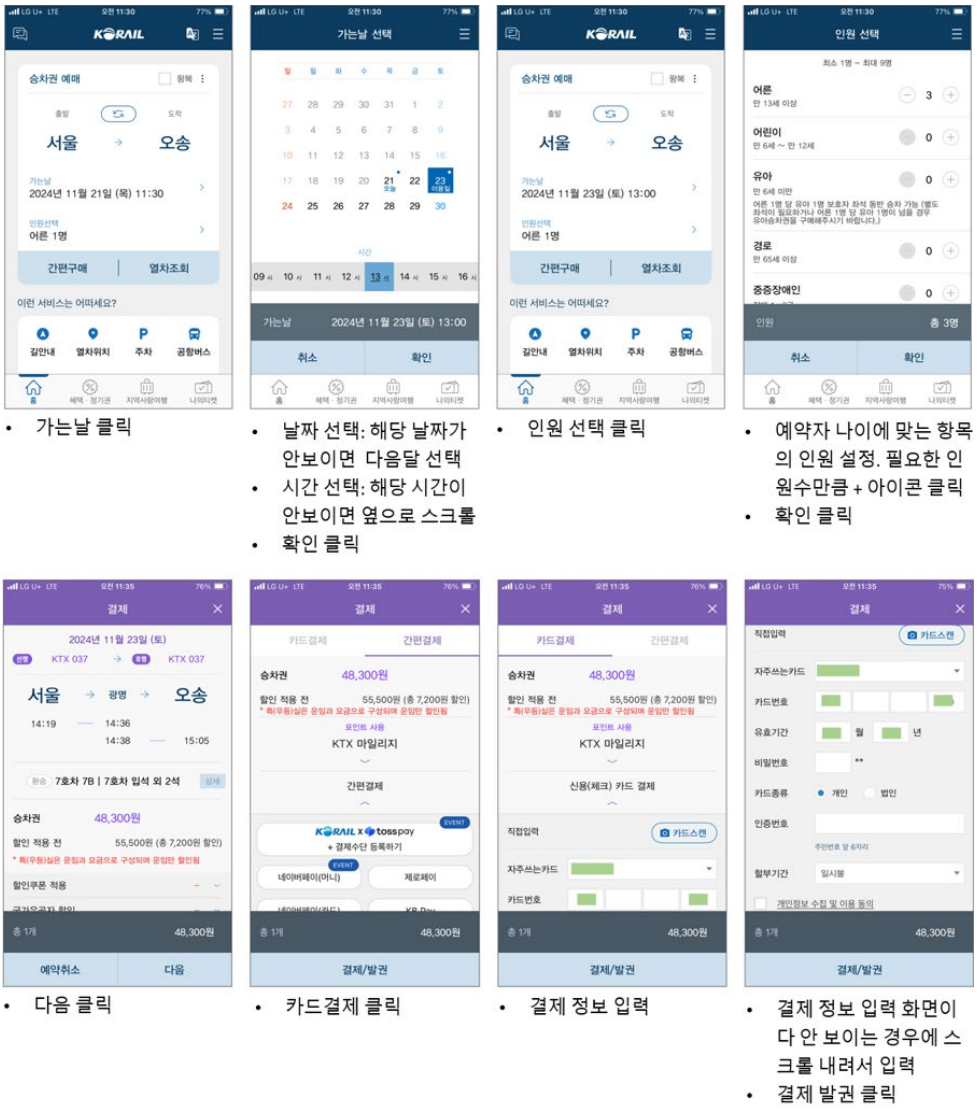


그림 6 코레일톡 앱을 이용한 기차표 예약에 필요한 Action 예

ADB는 안드로이드 기반의 장치와 통신하여 디버깅 등의 작업을 수행하는 Command Line Tool이며, 안드로이드 SDK에 포함되어 있다. ADB를 사용하기 위해서는 USB 또는 WiFi로 연결된 PC가 있어야 하므로 스마트폰만을 사용하는 실생활에는 적용할 수 없는 방식이다.

스마트폰이 독립적으로 동작하고 미리 정하지 않은 임의의 앱 사용을 자동화하기 위해서는 스마트폰에서 백그라운드로 동작하는 자동화 에이전트가 필요하다. 자동화 에이전트는 다음의 기능을 수행할 수 있어야 한다.

- 앱의 버튼이나 메뉴를 사람이 터치해서 실행하지 않고 해당 좌표 정보를 이용하여 터치한 것과 같은 효과를 낼 수 있어야 함. 가령, 버튼의 좌표가 (50, 50)이면 자동화 에이전트에서 click(50, 50) 함수를 호출 시 현재 화면에 보이는

앱의 (50, 50) 위치에 있는 버튼이 클릭되어야 함
- 현재 화면에 보이는 앱이 아니고 백그라운드 앱에서 스크린샷을 얻을 수 있어야 함

하지만, 복수의 안드로이드 보안 전문가 의견에 따르면 이러한 기능의 구현을 위해서 특별한 권한이 필요하다. 즉, 루팅을 하거나 제조사 권한이 필요하고, 안드로이드 OS 자체를 수정해야 할 수도 있다. 따라서, 현재 안드로이드 환경에서는 갤럭시와 같은 스마트폰에 자동화 에이전트를 탑재하는 것은 어렵다고 판단된다.

3. 사용 편의성

가. 처리 시간

AutoDroid[7] 논문에 따르면 Vicuna-7B Model을 스마트폰에서 실행 시 한 Step 실행에 30초 이상, 클라우드 GPT-4를 사용 시 20초 정도 소요된다. 처리 시간은 사용자 Instruction 완료에 필요한 Step의 수(즉, Episode 길이)에 비례하여 증가한다. 코레일톡 앱을 이용한 기차표 예약은 10개 이상의 Step을 거치므로 클라우드 GPT-4를 이용하더라도 3분 이상이 소요된다. 클라우드 기반 LLM은 On-Device LLM보다 빠르지만 비용이 발생할 수 있는 문제도 있다.

앱 UI가 복잡하면 UI의 텍스트 표현이 길어질 수 있고, 초기 화면에 모든 정보가 나타나지 않아서 스크롤 등을 통해서 전체 정보를 포함하는 화면을 재구성한다면 VLM에 입력되는 이미지 해상도 또한 높아져야 한다. 이와 같은 이유로 AITW[5] 데이터셋이 아니고 실사용 앱을 대상으로 시험하면 LLM에 입력되는 Token Size가 증가하여 처리 시간이 늘어날 가능성이 있다.

자동화 에이전트 사용자가 감내할 수 있는 처리 시간은 사용자 시험을 통해 확인해야 하지만 몇 분을 기다릴 수 있을 것으로 예상하기는 어렵다. 따라서, 경량 LLM을 사용하면서 정확도를 높일 수 있고, LLM에 입력되는 Token Size를 줄이는 연구가 필요하다.

나. 사용자 Instruction 입력

앱을 사용하기 전에 최종 목표와 중간 Step에서 필요한 정보까지 모두 포함하여 Instruction을 만들기는 어렵다. 고령인은 적절한 Instruction을 구성하는 데 특히 더 어려움을 겪을 것으로 예상되고, 텍스트를 타이핑해서 입력하는 자체가 진입 장벽이 될 수도 있다. 텍스트 입력뿐만 아니라 음성 입력 방식도 제공되어야 하며, 오타와 같은 오류를 자동 보정하는 기능도 필요할 것으로 예상된다.

그리고 필요한 서비스를 받기 위한 전체 과정에서 어려움을 겪기보다는 특정 Step에서만 도움이 필요할 가능성이 높다. 따라서 특정 Step에서만 자동화를 지원하는 기능도 필요하다.

Ⅴ. 결론

다양한 에이전트들이 제안되고 있지만 세부 동작 방식에 있어서 유사한 측면이 많다. 사용되는 Language Model이 성능에 큰 영향을 줄 것으로 예상되며, 최신 Language Model을 쉽게 교체 적용할 수 있는 구조로 자동화 에이전트를 개발하는 것이 중요하다. 그리고 현재 연구는 앱의 언어가 영어로 제한되어 있어서 한국어 앱에 대한 데이터셋과 이를 이용하여 학습한 에이전트의 개발이 필요하다.

지금까지 제안된 자동화 에이전트는 짧은 Episode에 대해서도 수십 초의 처리 시간이 걸리고, 앱 UI와 Action이 단순한 경우에도 Action Accuracy가 약 80% 수준이므로 사용자가 불편함 없이 사용하기에는 매우 부족한 것이 현실이다. 그리고 OS 보

안 이유로 스마트폰에서 독립적으로 동작하는 에이전트의 개발이 불가능한 것도 해결해야 하는 주요 이슈이다. 자동화 정확도를 높이고 처리 시간을 줄이면서 클라우드 LLM 사용으로 인한 비용 문제 등을 해소하여 스마트폰 사용자의 수요를 끌어낼 수 있으면 스마트폰 제조사가 자동화 에이전트가 실행될 수 있는 환경을 제공할 것으로 예상한다.

용어해설

스마트폰 앱 사용 자동화 앱 UI에서 필요한 메뉴의 선택을 스스로 판단하고 정보 입력에 필요한 사용자 개입을 최소화해서 사용자가 필요한 서비스를 누릴 수 있도록 보조하는 기술

약어 정리

ADB	Android Debug Bridge
AITW	Android in the Wild
API	Application Programming Interface
BC	Behavioural Cloning
LLM	Large Language Model
OCR	Optical Character Recognition
UI	User Interface
UTG	UI Transition Graph
VLM	Vision Language Model

참고문헌

[1] 강은나 외, "2023년도 노인실태조사," 한국보건사회연구원, 2023. https://www.mohw.go.kr/board.es?mid=a10411010100&bid=0019&act=view&list_no=1483359

[2] 최형임, 송인욱, "노인의 디지털 정보활용능력과 삶의 만족도의 관계에서 자기효능감의 매개효과 분석," 한국산학기술학회논문지, vol. 21, no. 6, 2020, pp. 246-255.

[3] 서울디지털재단, "2023년 서울시민 디지털역량 실태조사 주요 결과," 2024. https://sdf.seoul.kr/research-report/2471

[4] 서울디지털재단, "고령층 친화 디지털 접근성 표준," 2021. https://sdf.seoul.kr/research-report/1458

[5] C. Rawles et al., "Android in the Wild: A Large-Scale Dataset for Android Device Control," arXiv preprint, 2023. doi: 10.48550/arXiv.2307.10088.

[6] R. Anil et al., "Palm 2 technical report," arXiv preprint, 2023. doi: 10.48550/arXiv.2305.10403

[7] H. Wen et al., "AutoDroid: LLM-powered Task Automation in Android," in Proc. Int. Conf. Mobile Comput. Netw. (Washington D.C., DC, USA), Nov. 2024, pp. 543-557.

[8] W.L. Chiang et al., "Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality," May. 30th, 2023. https://lmsys.org/blog/2023-03-30-vicuna/

[9] Y. Song et al., "VisionTasker: Mobile Task Automation Using Vision Based UI Understanding and LLM Task Planning," arXiv preprint, 2024. doi: 10.1145/3654777.3676386

[10] J. Wang et al., "Mobile-Agent: Autonomous Multi-Modal Mobile Device Agent with Visual Perception," in Proc. Int. Conf. Learn. Representations Workshop on LLM Agents, (Vienna, Austria), May. 2024.

[11] J. Wang et al., "Mobile-Agent-v2: Mobile Device Operation Assistant with Effective Navigation via Multi-Agent Collaboration," in Proc. Conf. Neural Inf. Process. Syst., (Vancouver, Canada), Dec. 2024.

[12] Z. Zhang et al., "You Only Look at Screens: Multimodal Chain-of-Action Agents," in Proc. Assoc. Comp. Linguistics., (Bangkok, Thailand), Aug. 2024, pp. 3132-3149.

[13] W. Hong et al., "CogAgent: A Visual Language Model for GUI Agents," in Proc. Int. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., (Seattle, WA, USA), Jun. 2024, pp. 14281-14290.

[14] X. Ma et al., "CoCo-Agent: A Comprehensive Cognitive MLLM Agent for Smartphone GUI Automation," in Proc. Assoc. Comp. Linguistics., (Bangkok, Thailand), Aug. 2024, pp. 9097-9110.

[15] J. Li et al., "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models," in Proc. Int. Conf. Mach. Learn., (Honolulu, HI, USA), Jul. 2024, pp. 19730-19742.

[16] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," J. Mach. Learn. Res., vol. 21, no. 140, 2020, pp. 1-67.

[17] A. Radford et al., "Learning Transferable Visual Models from Natural Language Supervision," in Proc. Int. Conf. Mach. Learn, Jul. 2021, pp. 8748-8763.