NLP · LARGE LANGUAGE MODELS · META AI RESEARCH · arXiv:2302.04761

Toolformer

Language Models Can Teach Themselves to Use Tools

저자: Timo Schick · Jane Dwivedi-Yu · Roberto Dessì · Roberta Raileanu
Maria Lomeli · Luke Zettlemoyer · Nicola Cancedda · Thomas Scialom

핵심: 대형 언어 모델이 자가 감독(self-supervised)으로 외부 API 도구 사용법을 스스로 학습
→ 자신보다 25배 큰 GPT-3(175B)를 제로샷 평가에서 다수 과제에서 능가

5가지 외부 도구

```
QA("Where was Columbus founded?") → New Haven, CT
```

```
Calculator(400 / 1400) → 0.29
```

```
WikiSearch("Brown Act") → The Ralph M. Brown Act...
```

```
MT("sûreté nucléaire") → nuclear safety
```

```
Calendar() → Today is Monday, January 30, 2023
```

TABLE OF CONTENTS

발표 구성 — AS-IS · GAP · WHY NOW · TO-BE + 연구 설계 전 과정

01 AS-IS · Problem

LLM의 현재 한계 — 스케일의 역설, 5가지 내재적 약점

02 GAP

기존 해결책의 구조적 문제 — 인간 주석 의존, Task-Specific 한계

03 WHY NOW

왜 지금인가 — 3가지 Desiderata

04 TO-BE · Approach

자가 감독 3단계 파이프라인 · API 표현 수식 · 필터링 기준 수식

05 5가지 도구 설계

QA · Calculator · WikiSearch · MT · Calendar 구현 세부

06 연구 설계 · 데이터 수집

CCNet 코퍼스, 필터링 임계값별 데이터셋 규모 통계 (Table 2)

07 분석방법·실험조건·비교대상

측정 지표 전체 + 4개 내부·2개 외부 베이스라인 설계 의도

08 실험 결과 전 과제

LAMA·수학·QA·MLQA·시간·LM·Scaling Laws·Decoding Analysis

09 한계 · 결론

§7 원문 4가지 한계 + 후속 연구 방향

Source: Schick et al., 2023 — arXiv:2302.04761v1

01

AS-IS · CURRENT SITUATION

Problem — 현재 LLM의 역설적 한계

"Language models exhibit remarkable abilities... They also, paradoxically, struggle with basic functionality, such as arithmetic or factual lookup, where much simpler and smaller models excel." — Abstract

LLM의 5가지 구조적 약점 — 스케일링으로 해결되지 않는 문제들

논문 원문 (§1)

"However, all of these models have several inherent limitations that can at best be partially addressed by further scaling."

Schick et al., 2023, p.1

한계 ①

최신 정보 접근 불가

학습 컷오프 이후 사건·변경 사실 차단. 내재적 지식만으로 최신성 불가

Komeili et al., 2022

한계 ②

사실 환각 (Hallucination)

없는 사실을 자신감 있게 생성. 파라미터 지식의 불확실성 내재

Maynez et al., 2020; Ji et al., 2022

한계 ③

저자원 언어 이해 부족

훈련 데이터 희귀 언어에서 구조적 불균형. 다국어 과제 성능 저하

Lin et al., 2021

한계 ④

정밀 산술 계산 실패

간단한 사칙연산도 오류. 소형 계산기보다 못하는 역설적 상황

Patel et al., 2021

한계 ⑤

시간 감각 부재

현재 날짜·경과 시간·미래 일정 등 시간 기반 추론 근본적 불가

Dhingra et al., 2022

핵심 역설: GPT-3(175B) 같은 초거대 모델이 간단한 계산기나 달력보다 산술·시간 추론을 못하는 아이러니. 파라미터를 늘려도 이 문제들은 'at best partially addressed'될 뿐임. 근본 해결책 = 외부 도구 결합.

기존 해결책의 구조적 한계 — 왜 기존 방식이 부족한가

논문 원문 (§1)

"existing approaches either rely on large amounts of human annotations (Komeili et al., 2022; Thoppilan et al., 2022) or limit tool use to task-specific settings only (e.g., Gao et al., 2022; Parisi et al., 2022), hindering a more widespread adoption of tool use in LMs."

Schick et al., 2023, §1

기존 방법 A — 인간 주석 의존

대규모 Human Annotation 기반

사람이 직접 '이 맥락에서 이 API를 이렇게 써야 한다'는 가이드라인 주석 작성.

문제①: 막대한 비용 — 수백만 문서 주석 인력·시간
문제②: 판단 불일치 — 'what humans find useful may be different from what a model finds useful'
문제③: 확장성 없음 — 새 도구 추가 시 전체 재주석 필요

Komeili et al., 2022; Thoppilan et al. (LaMDA), 2022

기존 방법 B — Task-Specific 프롬프팅

태스크별 사전 지정 방식

Gao et al. (2022) 등: 특정 태스크에서 특정 도구를 쓰도록 few-shot 예시를 수동 구성.

문제①: 범용성 상실 — 사용자가 '어떤 도구를 어떻게 써야 하는지' 명시 필요
문제②: 도구 자율성 없음 — 'additional information is always provided, regardless of whether it is helpful'
문제③: 태스크 종속 — 한 태스크 프롬프트가 다른 태스크에 전이 안됨

Gao et al., 2022; Lazaridou et al., 2022; Yao et al., 2022

기존 방법 C — TALM (가장 유사한 선행 연구)

Self-supervised + 계산기·검색 한정

Parisi et al. (2022)의 TALM: 자가 감독 목표로 계산기·검색 학습.

한계: 'explores this only in settings where a model is finetuned on downstream tasks' → 다운스트림 태스크별 별도 파인튜닝 필요. 진정한 zero-shot 도구 사용 불가.

Toolformer 차별점: 자율적 도구 선택 + zero-shot 평가

Parisi et al., 2022 (TALM)

GAP 요약

① **자율성(Autonomy) 부재:** 모델이 스스로 '언제, 어떤 도구를, 어떻게' 쓸지 결정 불가. 인간 지정 필수. ② **Self-supervision 부재:** 도구 유용성을 모델 자신의 피드백(손실 감소)으로 판단하는 메커니즘 부재. Toolformer가 두 가지 동시 해결.

왜 지금인가 — Toolformer가 충족해야 할 3가지 Desiderata

논문 원문

"Therefore, we propose Toolformer, a model that learns to use tools in a novel way, which fulfills the following desiderata:"

Schick et al., 2023, §1

D1 Self-supervised 학습

""The use of tools should be learned in a self-supervised way without requiring large amounts of human annotations.""

이유: 인간이 유용하다고 생각하는 것과 모델이 실제 필요한 것이 다를 수 있으며, 주석 비용도 막대함.

Why Now①: In-context learning(Brown et al., 2020) 성숙 → 단 몇 개 예시로 대규모 데이터셋 자동 생성 가능. Self-supervised bootstrapping 시점 도래.

D2 범용성(Generality) 보존

""The LM should not lose any of its generality and should be able to decide for itself when and how to use which tool.""

이유: 특정 태스크에 종속되지 않고 어떤 맥락에서도 자율적으로 도구 사용 여부 판단해야 진정한 범용 에이전트.

Why Now②: 단순 파라미터 스케일링 한계 인식 확산. 수십조 파라미터 추가해도 산술 오류·환각 미해결. 도구 결합이 효율적 대안으로 부상.

D3 언어 모델링 능력 유지

""Our approach is agnostic of the dataset being used, we can apply it to the exact same dataset that was used to pretrain a model.""

이유: 도구 학습 후 기존 텍스트 생성 능력이 저하되면 실용성 없음. Trade-off 없이 능력 추가 필요.

Why Now③: 사전학습 데이터(CCNet)에 API 호출 삽입 → 동일 데이터로 파인튜닝 → perplexity 유지 가능. 이 방식이 LM 능력 보존과 도구 학습 동시 달성.

02

TO-BE · APPROACH & METHODOLOGY

Toolformer의 핵심 방법론 — self-supervised 3단계 파이프라인

Sample API Calls → Execute API Calls → Filter API Calls → Finetune LM on C*

API 표현 방식 + 3단계 파이프라인 — 텍스트 안에 API 호출 삽입

핵심 전제 (§2)

"Our aim is to equip a language model M with the ability to use different tools by means of API calls. We require that inputs and outputs for each API can be represented as text sequences. This allows seamless insertion of API calls into any given text."

Schick et al., §2

API 호출 표현 형식 — 튜플 c = (a_c, i_c)

`e(c)  = [ a_c(i_c) ]` ← 결과 없는 호출

`e(c,r) = [ a_c(i_c) → r ]` ← 결과 포함 호출

실제 토큰: `'[' ']' '→'` 를 `<API> </API>` → 대신 사용 (어휘 변경 없음)

실제 삽입 예시 (Figure 1 기반 — 4가지 도구)

```
...trademark of [QA("Who is the publisher?") → Massachusetts Medical Society] the MMS.
...400 (or [Calculator(400/1400) → 0.29] 29%) passed.
Spanish word for [MT("tortuga") → turtle] turtle.
[WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature...]
```

핵심 설계 결정: 동일 데이터셋 사용

"apart from inserted API calls the augmented dataset C contains the exact same texts as C" → 파인튜닝 M on C*는 C로 파인튜닝과 동일 콘텐츠에 노출 → LM 능력 보존 보장*

3단계 파이프라인 (FIGURE 2)

1 Sample API Calls

In-context learning으로 각 위치 i에서 API 호출 후보 생성.

확률: $p_i = p_M(\text{<API>} \mid P(x), x_{1:i-1})$

τ_s 이상인 위치 $I = \{i \mid p_i > \tau_s\}$ 선택
상위 k개 후보 $c^1_i \ldots c^m_i$ 샘플링

2 Execute API Calls

모든 후보를 실제 실행하여 응답 r_i 수신.
계산기: Python 스크립트
위키: BM25 retriever (KILT dump)
번역: NLLB 600M 파라미터
달력: 시스템 날짜 반환

3 Filter API Calls

채택 기준: $L^-_i - L^+_i \geq \tau_f$
(손실 감소가 τ_f 이상인 호출만 채택)

→ 원본 C에 삽입 → C* 구성
→ GPT-J (6.7B) 표준 LM 목표로 파인튜닝

필터링 기준 수식 완전 해설 — '언제 API 호출이 유용한가'의 수학적 정의

STEP 1: Sampling — 위치 및 후보 생성

각 위치 $i \in \{1,...,n\}$에서 API 시작 토큰 확률: $p_i = p_M(\text{<API>} \mid P(x), x_{1:i-1})$

$P(x)$는 도구별 in-context prompt. 임계값 τ_s 이상인 위치 $I = \{i \mid p_i > \tau_s\}$ 선택. 후보 k개 초과 시 상위 k개만 유지.

각 위치 $i \in I$에서 m개 API 호출 후보 $c^1_i,...,c^m_i$ 샘플링 후 실행.

STEP 2: Filtering — 가중 교차 엔트로피 손실 비교

$$L_i(z) = -\sum_{j=i}^{n} w_{j-i} \cdot \log p_M(x_j \mid z, x_{1:j-1})$$

가중치: $w_t = \max(0, 1-0.2 \cdot t) / \sum w_s$ → API 직후 토큰에 집중 (t=5 이상은 0)

$L^+_i = L_i(e(c_i, r_i))$ ← API 결과 포함 시 손실

$L^-_i = \min(L_i(\varepsilon), L_i(e(c_i, \varepsilon)))$ ← API 없을 때 최솟값 (호출 없음 vs 결과 없이 호출)

채택 조건: $L^-_i - L^+_i \geq \tau_f$ (기본값 $\tau_f = 1.0$)

"adding the API call and its result reduces the loss by at least τ_f, compared to not doing any API call or obtaining no result from it."

STEP 3: Model Finetuning 세부 하이퍼파라미터

배치 크기: 128 / 학습률: 1×10^{-5} / 웜업: 첫 10% 구간 선형 웜업

목적 함수: 표준 언어 모델링 (standard language modeling objective)

베이스 모델: GPT-J (Wang & Komatsuzaki, 2021) — 6.7B 파라미터

C* 구성: $x^* = x_{1:i-1}, e(c_i,r_i), x_{i:n}$ (다중 API 호출 반복 적용 → 전체 C에 대해 실행 → C*)

Inference: Top-k 디코딩 트릭 (k=10)

"We let the model start an API call not just when <API> is the most likely token, but whenever it is one of the k most likely tokens."

k=0(greedy): API 호출 0%

k=1: API 호출 40.3% (T-REx 기준)

k=3: API 호출 82.8%

k=10: API 호출 98.1%

루프 방지: 입력당 최대 1회 API 호출 제한

가중치 w_t 설계 의도

"to make sure that API calls happen close to where the information provided by the API is actually helpful"

API 바로 다음 토큰 예측에 가장 큰 가중치 → API 결과가 즉각 유용한 위치에서만 호출 발생 유도. t=5 이상: 가중치 0 (먼 미래 예측에는 영향 없음).

C* 구성의 핵심 특성

"finetuning M on C exposes it to the same content as finetuning on C"*

→ API 호출이 정확히 유용한 위치·입력으로 삽입 → LM이 피드백으로만 도구 사용법 자율 학습

Source: Schick et al., 2023. §2 Approach — arXiv:2302.04761

5가지 외부 API — 설계 제약 조건 + 구현 세부 + 채택률

도구 선정 기준 (§3)

"The only constraints we impose on these tools is that (i) both their inputs and outputs can be represented as text sequences, and (ii) we can obtain a few demonstrations of their intended use."

Tool 1 · QA

Question Answering

구현: Atlas (Izacard et al., 2022) — retrieval-augmented LM finetuned on NQ (Kwiatkowski et al.)
입력: 자연어 질문 문자열
출력: 짧은 사실형 답변 (factoid answer)
평가 시: LAMA에서 WikiSearch 차단 (Wikipedia 기반 태스크 → 불공정 방지)
채택률: LAMA 98.1% / 수학 비활성화

```
QA("Where was Columbus found?") → New Haven, CT
```

Tool 2 · Calculator

Calculator

구현: Python 스크립트 (사칙연산 +−×÷만 지원)
출력: 소수점 2자리 반올림 결과
휴리스틱: 숫자 3개 이상 포함 텍스트에서만 API 생성 시도
이유: 불필요한 계산기 호출 사전 차단으로 효율화
채택률: 수학 과제에서 97.9%

```
Calculator(27 + 4 * 2) → 35
```

Tool 3 · WikiSearch

Wikipedia Search

구현: BM25 retriever (Robertson, 1995; Baeza-Yates, 1999)
인덱스: KILT Wikipedia dump (Petroni et al., 2021)
출력: 검색어 관련 짧은 텍스트 스니펫
한계: 단발성 쿼리 → 재검색 불가. 모델이 직접 관련 부분 추출 필요
채택률: WebQS/NQ에서 99.3%

```
WikiSearch("Fishing Reel Types") → Spin fishing > ...
```

Tool 4 · MT

Machine Translation

구현: NLLB 600M 파라미터 (Costa-jussà et al., 2022)
지원: 200개 언어 (저자원 포함)
소스 언어: fast-Text classifier 자동 감지
타깃 언어: 항상 영어로 고정
채택률: MLQA 언어별 63.8%~94.9% (Hindi 예외 7.3%)

```
MT("sûreté nucléaire") → nuclear safety
```

Tool 5 · Calendar

Calendar

구현: 입력값 없이 호출 → 시스템 현재 날짜 반환
특이점: 5가지 중 유일하게 i_c 없음 — Calendar() 형태
역할: 'provides temporal context for predictions that require some awareness of time'
채택률: DATESET 54.8% / TEMPLAMA 불과 0.2% (자율 구별!)

```
Calendar() → Today is Monday, January 30, 2023
```

연구 설계 · 데이터 수집 전 과정 · 데이터셋 규모 통계

연구 설계 유형

알고리즘 구현 + 정량 실험 연구 (Algorithm Design + Empirical Evaluation). 인간 레이블 없이 모델 자체 피드백(loss)으로 학습 데이터 자동 생성하는 자가 감독 학습 알고리즘을 설계·구현하고, 기존 베이스라인 대비 정량적 성능 비교 수행.

기본 텍스트 코퍼스 — CCNet (Wenzek et al., 2020) 서브셋

"Throughout all of our experiments, we use a subset of CCNet as our language modeling dataset C."

CCNet: Common Crawl을 언어 감지 및 품질 필터링한 대규모 웹 코퍼스.
각 도구별로 API 호출이 유용할 가능성이 높은 텍스트를 선별하는 휴리스틱 필터 추가 적용:
· Calculator: 숫자 3개 이상 포함 텍스트만 대상
· 나머지 도구: 도구별 Appendix A 기준 적용
필터링 임계값 τ_s, τ_f는 도구별 개별 설정 → 충분한 학습 예시 확보 목표.

In-context Learning Prompt 설계 (Figure 3 기반)

각 API별로 소수의 예시가 포함된 prompt P(x) 작성. QA 도구 prompt 예시(Figure 3):

"Your task is to add calls to a Question Answering API to a piece of text. The questions should help you get information required to complete the text. You can call the API by writing '[QA(question)]'."

Input: "Joe Biden was born in Scranton, Pennsylvania."
Output: "Joe Biden was born in [QA("Where was Joe Biden born?")] Scranton, [QA("In which state is Scranton?")] Pennsylvania."

이 prompt로 GPT-J가 대규모 CCNet 텍스트에 API 후보를 자동 삽입 → 실행 → 필터링.

Table 2 — τ_f별 데이터 규모

필터링 임계값 τ_f 값에 따라 최종 학습에 사용되는 API 호출 예시 수:

API	τ_f=0.5	τ_f=1.0	τ_f=2.0
Question Answering	51,987	18,526	5,135
Wikipedia Search	**207,241**	**60,974**	**13,944**
Calculator	3,680	994	138
Calendar	61,811	20,587	3,007
Machine Translation	3,156	1,034	229

주목: Calculator 994개 — '100만+ 문서 처리 → 수천 개 유용 호출만' (§7 샘플 비효율성의 대표 사례)

WikiSearch 60,974개 가장 많음 — 웹 텍스트 특성상 사실 확인·배경 정보 빈번. BM25가 다양한 쿼리에서 어느 정도 유용 결과 반환 → 채택률 높음.

Source: Schick et al., 2023. §4.1, Table 2, Appendix A — arXiv:2302.04761

분석 방법 — 과제별 측정 지표 · 평가 방식 전체 목록

평가 방식 공통 설정

"we consider a prompted zero-shot setup — i.e., models are instructed to solve each task in natural language, but we do not provide any in-context examples. This is in contrast to prior work on tool use where models are provided with dataset-specific examples."

Schick et al., §4.2

지표① LAMA

SQuAD / Google-RE / T-REx

원래 마스킹 LM용으로 설계된 LAMA를 left-to-right LM 평가에 적용.

평가 기준: '정답이 예측 상위 5단어 안에 포함' (완전 일치보다 완화)

이유: 다양한 토크나이제이션 처리, 단일 단어만 요구하지 않음을 감안

Toolformer 주의: LAMA가 Wikipedia 기반 → WikiSearch API 차단 (불공정 유리 방지)

지표② 수학 데이터셋

ASDiv / SVAMP / MAWPS

ASDiv (Miao et al., 2020): 다양한 수학 문장제

SVAMP (Patel et al., 2021): 구조 변형에 강건한 수학 평가

MAWPS (Koncel-Kedziorski et al., 2016): 산수 문장제

평가 기준: '예측 텍스트에서 첫 번째 등장 숫자 = 정답'

예외: '5+3=8' 형태 → '=' 이후 첫 숫자

지표③ QA

WebQS / NQ / TriviaQA / MLQA

WebQS (Berant et al., 2013): Freebase 기반 웹 질문

NQ (Kwiatkowski et al., 2019): Natural Questions

TriviaQA (Joshi et al., 2017): 잡학 지식 QA

MLQA (Lewis et al., 2019): 스페인·독·힌·베·중·아랍어

평가 기준: '예측 첫 20단어 내 정답 포함' (MLQA: 10단어)

NQ 평가 시 QA 도구 비활성화 (NQ로 학습된 도구 → 공정성)

지표④ 시간 추론

TEMPLAMA / DATESET

TEMPLAMA (Dhingra et al., 2022): 시간에 따라 변하는 사실 추론

('Cristiano Ronaldo plays for ___')

DATESET: 저자 직접 생성. Wikidata 기반 템플릿. 2010~2020년 사실+날짜 계산

('What day of the week was it 30 days ago?')

평가 기준: LAMA와 동일 (상위 5단어 내 정답 포함)

지표⑤ 언어 모델링

Perplexity — WikiText & CCNet

목적: LM 능력 손상 없음을 증명 (Desideratum D3 검증)

WikiText (Merity et al., 2017): 표준 LM 벤치마크

CCNet 서브셋: 10,000개 문서 무작위 (학습 미사용)

주의: API 활성화 시 perplexity 계산에 marginalization 필요 → 불가

→ Toolformer (disabled)만 평가 (API 토큰 확률=0)

Source: Schick et al., 2023. §4.2, §4.3 — arXiv:2302.04761

실험 조건 통제 · 4개 내부 + 2개 외부 베이스라인 — 설계 의도

통제 변수

통 제 ①

모델 아키텍처·크기 고정

모두 GPT-J(6.7B) 기반 고정. 순수한 도구 학습 방법론의 효과만 격리 측정. 아키텍처·파라미터 차이가 없으므로 성능 차이 = 학습 방식의 차이.

통 제 ②

Zero-shot 평가 환경

"We choose the more challenging zero-shot setup as we are interested in seeing whether Toolformer works in precisely those cases where a user does not specify in advance which tools should be used."

통 제 ③

태스크별 API 수동 차단

LAMA → WikiSearch 차단 (Wikipedia 기반)
NQ 평가 → QA 도구 차단 (NQ 학습 도구 → 불공정)
목적: 'this would make solving the tasks trivial, especially given that the underlying QA system was finetuned on NQ.'

베이스라인 모델 — 설계 의도

GPT-J 6.7B 내부

파인튜닝 전혀 없는 순수 기본 모델. 모든 성능 향상의 하한선.

GPT-J + CC 6.7B 내부

API 없이 CCNet으로만 추가 파인튜닝. 데이터 분포 변화(distribution shift) 효과만 격리. C*와 C의 차이 = API 호출 자체 기여.

Toolformer (disabled) 6.7B 핵심

C*로 학습했지만 추론 시 <API> 토큰 확률 강제로 0 설정. 도구 직접 사용 효과 vs 도구 데이터로 학습하는 지식 내재화 효과를 분리.

Toolformer 6.7B 제안

제안 모델. k=10 Top-k 디코딩으로 API 호출 완전 활성화. 모든 비교의 기준.

OPT 66B 외부

Toolformer보다 10× 큰 외부 베이스라인 (Zhang et al., 2022). 파라미터 체급 차이가 도구 사용보다 중요한지 검증.

핵심 비교군 설계 의도 — Toolformer vs Toolformer(disabled): 동일 C*로 동일하게 학습, 유일한 차이 = 추론 시 도구 사용 여부. **만약 disabled ≈ GPT-J → 도구 사용 자체가 원인.** 만약 disabled > GPT-J → 도구 데이터로 학습하는 것 자체가 내부 지식 향상. 실험 결과: 두 효과 모두 존재.

Source: Schick et al., 2023. §4.1 Experimental Setup — arXiv:2302.04761

LAMA 사실 검증 결과 — 동급 모델 압도 + GPT-3(175B) 능가

33.8
LAMA SQuAD (Toolformer)
vs GPT-3 26.8
↑+7.0p

11.5
LAMA Google-RE (Toolformer)
vs GPT-3 7.0
↑+4.5p

53.5
LAMA T-REx (Toolformer)
vs GPT-3 39.8
↑+13.7p

98.1%
QA 도구 채택률 (k=10)
거의 항상 QA 도구 사용

Table 3: LAMA 결과

Model	SQuAD	G-RE	T-REx
GPT-J	**17.8**	**4.9**	**31.9**
GPT-J+CC	**19.2**	**5.6**	**33.2**
TF(disabled)	**22.1**	**6.3**	**34.9**
Toolformer	**33.8**	**11.5**	**53.5**
—— 외부 대형 모델 ——			
OPT (66B)	21.6	2.9	30.1
OPT (66B)	**21.6**	**2.9**	**30.1**
GPT-3 (175B)	26.8	7.0	39.8
GPT-3 (175B)	**26.8**	**7.0**	**39.8**

해석: disabled > GPT-J+CC
disabled가 GPT-J+CC보다 모두 높음 → 단순 CCNet 파인튜닝이 아니라 API 호출 데이터 자체가 지식 내재화에 기여. 도구 사용 데이터를 보는 것만으로도 사실 지식 증가.

해석: OPT(66B)가 GPT-J보다 낮은 이유
OPT 21.6 < GPT-J 17.8보다는 높지만 Toolformer 33.8보다 훨씬 낮음. 파라미터 체급(10×)이 도구 사용 효과를 압도하지 못한다는 명확한 증거. 학습 데이터·아키텍처·파인튜닝 방식이 복합 영향.

LAMA 평가 세부: 마스킹 → 생성 모델 전환
마스크 토큰이 마지막 위치가 아닌 예시 제외. 완전 일치 대신 '상위 5단어 내 포함' 기준 사용 (토크나이제이션 처리). 논문 내 일관 적용으로 공정 비교.

Source: Schick et al., 2023. §4.2.1, Table 3 — arXiv:2302.04761

수학 추론 + QA 결과 — 계산기로 GPT-3 2배 압도 / 검색으로 동급 압도 but GPT-3 미달

수학 데이터셋 (Table 4)

40.4% ASDiv — vs GPT-3 14.0%

29.4% SVAMP — vs GPT-3 10.0%

44.0% MAWPS — vs GPT-3 19.8%

Model	ASDiv	SVAMP	MAWPS
GPT-J	7.5	5.2	9.9
GPT-J+CC	9.6	5.0	9.3
TF(disabled)	14.8	6.3	15.0
Toolformer	**40.4**	**29.4**	**44.0**
OPT (66B)	6.0	4.9	7.9
GPT-3 (175B)	14.0	10.0	19.8

지식 내재화 + 도구 직접 효과 분리

disabled 14.8 vs GPT-J 7.5 = +7.3p (내재화 효과)

Toolformer 40.4 vs disabled 14.8 = +25.6p (도구 직접 효과)

'the model is finetuned on many examples of API calls and their results, improving its own mathematical capabilities.' 도구 데이터 보는 것 자체가 수학 능력 향상.

QA 데이터셋 (Table 5)

26.3 WebQS — vs GPT-3 29.0

17.7 NQ — vs GPT-3 22.6

48.8 TriviaQA — vs GPT-3 65.9

Model	WebQS	NQ	TriviaQA
GPT-J	18.5	12.8	43.9
GPT-J+CC	18.4	12.2	45.6
TF(disabled)	18.9	12.6	46.7
Toolformer	**26.3**	**17.7**	**48.8**
OPT (66B)	18.6	11.4	45.7
GPT-3 (175B)	29.0	22.6	65.9

원인 분석 — WikiSearch 99.3% 활용했지만

BM25 한계: 쿼리와 맞지 않는 결과 반환 다수

단발성 호출: 결과 부적합해도 재쿼리 불가, 결과 탐색 불가

GPT-3: 방대한 내재 지식 → 검색 없이도 높은 성능 (TriviaQA 65.9)

'adding this functionality is an exciting direction for future work.'

다국어 QA + 시간 추론 + 언어 모델링 — MT 효과·Calendar 선택·Perplexity 보존

MLQA — 다국어 QA (Table 6)

Model	Es	De	Hi	Vi	Zh	Ar
GPT-J	15.2	16.5	1.3	8.2	18.2	8.2
GPT-J+CC	15.7	14.9	0.5	8.3	13.7	4.6
Toolformer	**20.6**	**13.5**	**1.4**	**10.6**	**16.8**	**3.7**
OPT 66B	0.3	0.1	1.1	0.2	0.7	0.1
GPT-3 175B	3.4	1.1	0.1	1.7	17.7	0.1

MT 도구: 언어별 63.8~94.9% 채택 (Hi 예외 7.3%)
분포 변화(CCNet)로 일부 언어에서 GPT-J 열세
OPT·GPT-3: 영어 답변 지시 실패로 극히 저조

시간 추론 (Table 7)

Model	TEMPLAMA	DATESET
GPT-J	13.7	3.9
GPT-J+CC	12.9	2.9
TF(disabled)	12.7	5.9
Toolformer	**16.3**	**27.3**

Calendar 선택적 사용 — 자율 구별력

TEMPLAMA: Calendar 채택 0.2%
이유: named entity 기반 사실 추론 → 날짜보다 QA·WikiSearch 가 더 유용. 모델이 스스로 파악.

DATESET: Calendar 채택 54.8%
날짜 계산 문제 → 'considerable improvement can be fully accredited to the calendar tool.'

GPT-3 DATESET 0.8% — 현재 날짜 정보가 없어 근본적으로 불가.

언어 모델링 Perplexity (Table 8)

Model	WikiText	CCNet
GPT-J	9.9	10.6
GPT-J+CC	10.3	10.5
TF(disabled)	**10.3**	**10.5**

§4.3
"Adding API calls comes without a cost in terms of perplexity for language modeling without any API calls."

Desideratum D3 검증

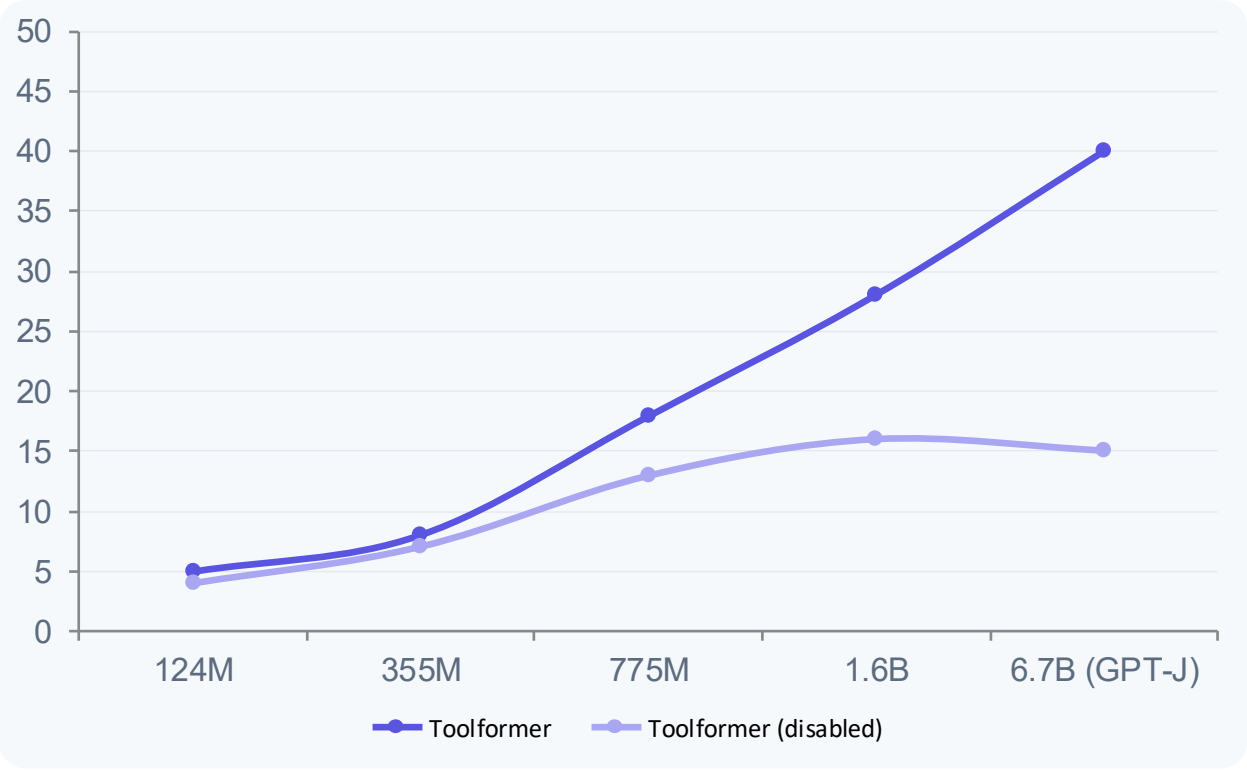
GPT-J+CC와 TF(disabled)가 동일 perplexity
이유: C*는 C와 동일 텍스트 + API 호출만 추가
→ LM 손실 계산 시 원본 텍스트 예측 능력 유지

WikiText 소폭 상승(9.9→10.3):
CCNet(랜덤 서브셋)과 WikiText 분포 차이 때문
(도구 때문이 아님 — GPT-J+CC도 동일 현상)

Desideratum D3 달성: Trade-off 없이 도구 학습!

MLQA 결론: MT 도구 사용이 모든 언어에서 일관 성능 향상 but CCNet 분포 변화로 일부 언어에서 원래 GPT-J 열세. OPT·GPT-3는 영어 지시 실패로 극히 저조 (GPT-J가 EuroParl 등 더 많은 다국어 데이터 학습). **Calendar 결론:** 동일 '시간' 관련 태스크인데도 모델이 TEMPLAMA(사실 추론)와 DATESET(날짜 계산)을 스스로 구별 → 자율적 도구 선택 능력 확인.

Scaling Laws — 도구 활용 능력은 언제 출현하나? 최소 모델 규모 조건



수학 벤치마크 성능 (ASDiv 기준, 개념적 재구성)

실용적 시사점

최소 ~775M 파라미터 이상에서만 Toolformer 방법론이 효과적. 소형 모델에는 이 방식 적용 어려움 — in-context learning 능력이 부족하면 초기 API 호출 샘플 자체를 제대로 생성하지 못하여 C* 구성 불가.

§4.4 SCALING LAWS 원문

"Figure 4 shows that the ability to leverage the provided tools only emerges at around 775M parameters: smaller models achieve similar performance both with and without tools. An exception to this is the Wikipedia search engine used mostly for QA benchmarks; we hypothesize that this is because the API is comparably easy to use."

Schick et al., 2023, §4.4

실험 설계: GPT-2 Family + GPT-J

적용 모델: 124M / 355M / 775M / 1.6B / 6.7B (5가지 규모)

사용 도구: 전체 5개 중 3개 — QA, 계산기, Wikipedia 검색

동일 실험 설정 (§4.1) 적용 — zero-shot 평가

핵심 발견: 775M 파라미터 임계점

124M~355M: 도구 유무 성능 거의 동일 → 유용한 API 쿼리 생성 실패

775M 이상: 격차 뚜렷 → In-context learning 성숙으로 도구 사용 가능

WikiSearch 예외: 소형 모델에서도 효과 ('API is comparably easy to use')

큰 모델에서도 도구 격차 유지

"there remains a large gap between predictions with and without API calls even for our biggest model." → 스케일 업 + 도구 사용은 대체 관계가 아닌 보완 관계.

Source: Schick et al., 2023. §4.4, Figure 4 — arXiv:2302.04761

디코딩 전략 분석 + 데이터 품질 검증 — 필터링 기준의 실제 작동 확인

디코딩 전략: k값별 성능 (Table 9 — T-REx 기준)

All=전체 / AC=API 호출한 예시 / NC=호출 안 한 예시 / %=API 호출 비율

k	All	AC	NC	API%
0	34.9	—	34.9	0.0%
1	47.8	53.0	44.3	40.3%
3	52.9	58.0	29.0	82.8%
10	**53.5**	**54.0**	**22.5**	**98.1%**

K=1의 자기 교정 능력 (§5)

"For k=1, the model is calibrated to some extent: It decides to call APIs for examples that it would perform particularly badly on without making API calls. This can be seen from the fact that performance where it decides not to make an API call (44.3) is higher than average performance if no API calls are made at all (34.9)."

Schick et al., §5

k값에 따른 Trade-off

k가 커질수록:

→ API 호출 비율 증가 (0%→98.1%)

→ All 성능 향상 (34.9→53.5)

→ NC 성능 하락 (34.9→22.5): 잘 알아도 API 먼저 시도

→ 'this calibration is lost for higher values of k'

k=10 선택 이유: All 성능 최대화 + 모델이 가능한 한 도구 활용하게 유도

Source: Schick et al., 2023. §5 Analysis, Tables 9–10 — arXiv:2302.04761

데이터 품질 — L^-_i–L^+_i 점수 기준 (Table 10 발췌)

점수 높을수록 실제 유용. 필터링 기준(τ_f=1.0)의 타당성을 질적 검증.

5.49

`WikiSearch("War memorial Flodden") → Battle of Flodden > The stained-glass Flodden Window was constructed by Sir Richard Assheton...`

매우 유용 — 희귀 역사 정보 정확 검색

2.08

`QA("Approximate length of the Nile?") → 6,853 km`

유용 — 수치 사실 정확 답변

0.92

`WikiSearch("Fast train success") → Fast Train > peaked at #23 on Canadian CHUM chart...`

채택됐지만 문맥 무관 음악 정보 — 노이즈 (τ_f=1.0 아래라 실제 제외)

-0.02

`Calculator(85 / 23) → 3.70` (문맥: `85 patients 23%` → 나눗셈이 아님)

채택 안됨 — 문맥상 계산 불필요

-1.23

`QA("Who was last time I was with?") → The Last Time`

채택 안됨 — 개인 맥락에 QA 호출 무의미

'some amount of noise in the API calls that are not filtered can be useful as it forces the model to not always blindly follow the results.' — §5

03

§7 LIMITATIONS · §8 CONCLUSION

한계 · 결론

"While our approach enables LMs to learn how to use a variety of tools in a self-supervised way, there are some clear limitations to what can be achieved with our method in its current form." — §7

4가지 구조적 한계

L1 · 도구 연쇄 불가

No Tool Chaining

"One such limitation is the inability of Toolformer to use tools in a chain (i.e., using the output of one tool as an input for another tool). This is due to the fact that API calls for each tool are generated independently; as a consequence, there are no examples of chained tool use in the finetuning dataset."

→ 영향

다단계 추론 불가. 달력→QA 연쇄(TEMPLAMA 저조) 등 근본적 병목.

→ 후속 연구

ReAct (Yao et al.), Chain-of-Thought + Tool 결합 연구로 발전

L2 · 비상호작용적 사용

Non-interactive Tool Use

"Our current approach also does not allow the LM to use a tool in an interactive way – especially for tools such as search engines, that could potentially return hundreds of different results, enabling a LM to browse through these results or to refine its search query."

→ 영향

BM25 결과 부적합해도 재쿼리 불가 → QA에서 GPT-3 열세의 주요 원인.

→ 후속 연구

WebGPT (Nakano et al.), 인터랙티브 검색 에이전트 연구로 발전

L3 · 입력 민감성

Input Sensitivity

"Beyond this, we found models trained with Toolformer to often be sensitive to the exact wording of their input when deciding whether or not to call an API; this is perhaps unsurprising given that LMs are known to be very sensitive to the prompt they are provided with."

→ 영향

동일 사실 다른 표현으로 질문 시 API 호출 여부 달라짐. 일관성 부재.

→ 후속 연구

Calibration 연구, Prompt Robustness 개선 방향

L4 · 샘플 비효율+비용 무감각

Sample Inefficiency & Cost Unawareness

"Our method is also very sample-inefficient; for example, processing more than a million documents results in only a few thousand examples of useful calls to the calculator API. Toolformer currently does not take into account the tool-dependent, computational cost incurred from making an API call."

→ 영향

계산기 994개, 번역 1,034개 극소 데이터. 전처리 비용 막대. API 비용 무시.

→ 후속 연구

Bootstrapping 반복 적용 (§7 제안), Cost-aware RL 기반 의사결정

Source: Schick et al., 2023. §7 Limitations — arXiv:2302.04761

결론 — 무엇을 증명했고, 어디로 이어졌는가

§8 CONCLUSION 원문

"We have introduced Toolformer, a language model that learns in a self-supervised way how to use different tools such as search engines, calculators, and translation systems via simple API calls. This is done by finetuning on a large number of sampled API calls that are filtered based on whether they reduce perplexity on future tokens. Toolformer considerably improves zero-shot performance of a 6.7B parameter GPT-J model, enabling it to even outperform a much larger GPT-3 model on a range of different downstream tasks."

Schick et al., 2023, §8

AS-IS → TO-BE 달성

패러다임 전환 증명

인간 주석 없이 자가 감독만으로 도구 사용 학습 가능 증명.
5가지 LLM 내재 한계를 외부 도구로 보완.
3가지 Desiderata 모두 충족.
Perplexity Trade-off 없음 (D3 검증).

수치 요약

· 수학(ASDiv): GPT-3 14.0 → TF 40.4

· 사실(T-REx): GPT-3 39.8 → TF 53.5

· 시간(DATESET): GPT-3 0.8 → TF 27.3

· Perplexity: 보존 (CCNet 10.5→10.5)

연구 기여의 핵심

효율성의 증명

6.7B가 175B를 이긴다 — 도구 사용이 파라미터 스케일링보다 특정 약점 해결에 훨씬 효율적.

핵심 설계의 단순함: API 호출 텍스트 삽입 → 손실 감소 시 채택 → 파인튜닝. 특수 아키텍처 없이 표준 LM objective만으로 구현.

수치 요약

· OPT(66B) 대비 수학에서 6.7× 이상 성능

· GPT-3(175B) 대비 T-REx +13.7p

· 계산기 97.9% / WikiSearch 99.3%

· 자율적 도구 선택 (Calendar 자율 구별)

후속 연구 영향

AI 에이전트의 초석

Function Calling (OpenAI API)
ReAct (Yao et al., 2023)
Gorilla (Patil et al., 2023)
ToolLLM (Qin et al., 2023)
LangChain / LlamaIndex Tool Use
OpenAI Plugins / GPT-4 Tools

현재 AI 에이전트 생태계의 핵심 선행 연구.

수치 요약

· 남은 과제: Tool Chaining

· Interactive Query 재수정

· Cost-aware 의사결정

· 더 작은 모델에서의 적용

Source: Schick et al., 2023. §8 Conclusion, §7 Limitations — arXiv:2302.04761v1, 9 Feb 2023

ReAct

SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS

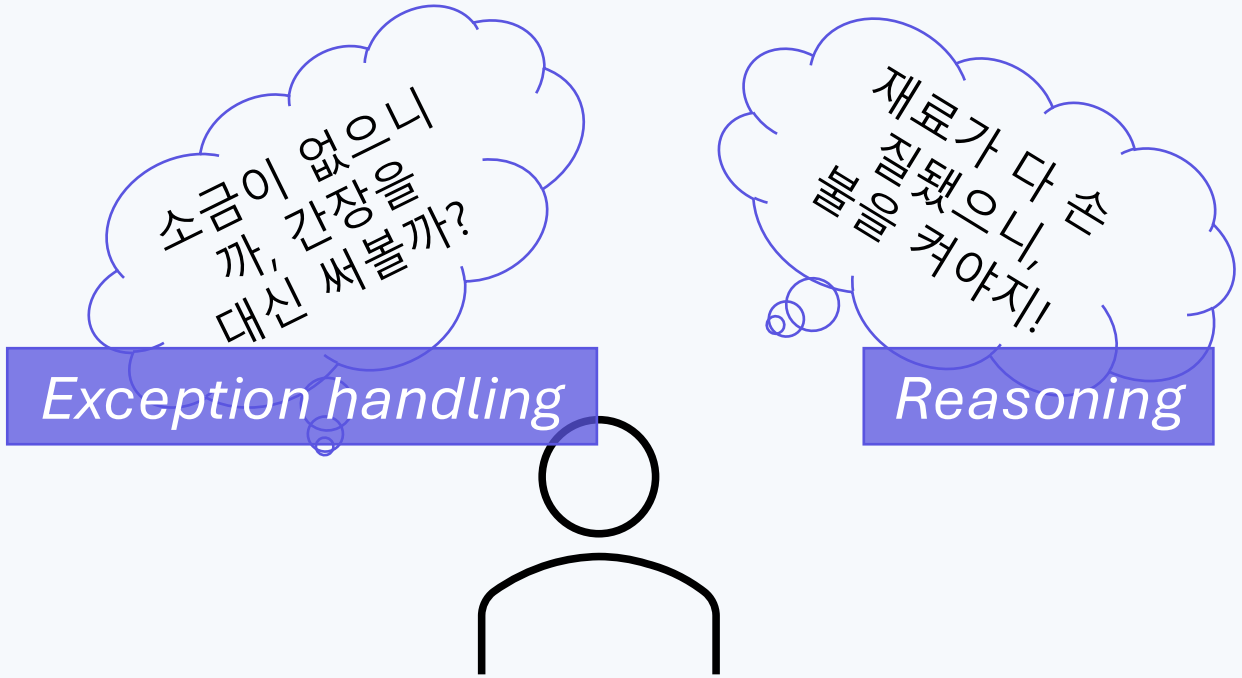
Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, Yuan Cao

Background

Human Intelligence의 key feature

: Task-oriented action + verbal reasoning

e.g. cooking



외부 환경으로부터 새로운 정보 습득 & 현재 진행 상황을 추적하고 계획을 수정

기존 연구의 한계점

CoT(Chain-Of-Thought)

: reasoning only

: 모델 내부 지식에만 의존하는 블랙 박스 형태

→ fact hallucination 발생 가능성

Action-Only Plan

: acting only

: 단순한 명령은 생성할 수 있지만, 고차원적인 목표 이해나 상태를 추적하는 능력이 부족

→ 복잡한 환경에서 무한 loop에 빠질 가능성

REACT: REasoning + ACTing

: LLM이 verbal reasoning trace와 action을 모두 generate하게끔 prompt

: 주어진 task를 해결하기 위해 agent가 environment와 interact하는 상황 가정

- time step t에서 agent는 observation $o_t \in O$를 receive하고, action $a_t \in A$를 행함
- 이때 agent의 context는 $c_t = (o_1, a_1, \ldots, o_{t-1}, a_{t-1}, o_t)$

Key Idea
Action space를 $\hat{A} = A \cup L$로 augment
L은 language space

REACT: REasoning + ACTing

Cf)
Thought: environment에 직접 영향을 주지 않는 자유로운 추론
Action: 외부 API 호출처럼 환경과 상호작용하는 동작
Observation: action의 결과로 environment가 돌려주는 feedback

- Language space

: language space 안의 action $\hat{a}_t \in L$ (thought/reasoning trace)은 external environment에 영향을 미치지 않으며, 따라서 observation feedback이 없음

: 대신 $\hat{a}_t$ 는 유용한 정보를 목표로 함

: 근데 language space가 unlimited하다는 문제

→ language model 고정, few-shot in-context example로 prompt

REACT

Question	What is the elevation range for the area that the eastern sector of the Colorado orogeny extends into?
Thought 1	I need to search Colorado orogeny, find the area that the eastern sector of the Colorado orogeny extends into, then find the elevation range of the area.
Action 1	Search[Colorado orogeny]
Observation 1	The Colorado orogeny was an episode of mountain building (an orogeny) in Colorado and surrounding areas.
Thought 2	It does not mention the eastern sector. So I need to look up eastern sector.
Action 2	Lookup[eastern sector]
Observation 2	(Result 1 / 1) The eastern sector extends into the High Plains and is called the Central Plains orogeny.
Thought 3	The eastern sector of Colorado orogeny extends into the High Plains. So I need to search High Plains and find its elevation range.
Action 3	Search[High Plains]
Observation 3	High Plains refers to one of two distinct land regions
Thought 4	I need to instead search High Plains (United States).
Action 4	Search[High Plains (United States)]

: prompt 예시

REACT: REasoning + ACTing

1. Reasoning이 중요한 task

: thought와 action을 번갈아 generate

→ task solving trajectory에 여러 thought-action-observation step이 포함됨

2. Decision making이 중요한 task

: 이 경우 잠재적으로 많은 수의 acting이 내재되어 있음

→ thought는 sparse하게 trajectory에서 등장

→ language model이 알아서 thought의 asynchronous한 occurrence를 결정하게 함

Benchmark

Knowledge-Intensive

: HotpotQA, FEVER

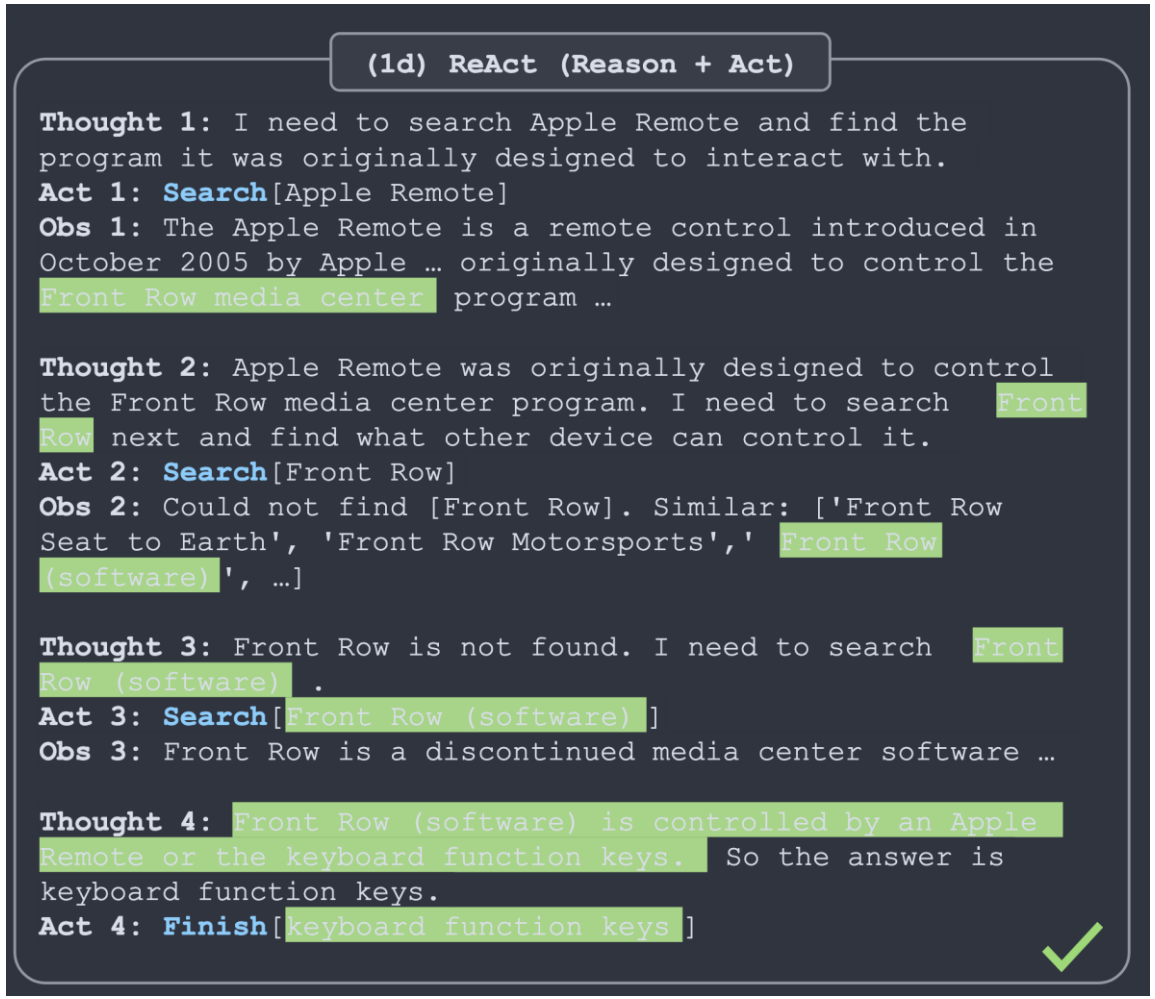
: 위키피디아 API를 통해 multi-hop 질의 응답 및 사실 검증

Decision making

: ALFWorld, WebShop

: 가상 공간 내 물체 조작 및 실제 웹 쇼핑몰 내비게이션 테스크

Knowledge-Intensive Reasoning Task



: ReAct는 위키피디아 API와 interact하며 reasoning을 support하는 information을 retrieve함

: 위키피디아 API

1. search[entity] : entity에 대응하는 page에서 5 sentence retrieve
2. lookup[string] : ctrl + F
3. finish[answer]

→ 실제 사람이 위키피디아와 interact하는 방식을 묘사

Knowledge-Intensive Reasoning Task: Result

1. ReAct > Act
2. ReAct vs CoT

Hallucination

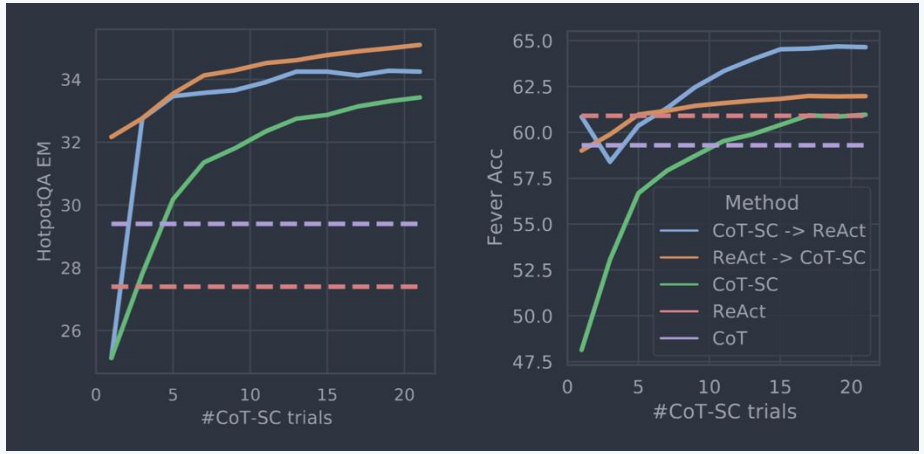
	Type	Definition	ReAct	CoT
Success	True positive	Correct reasoning trace and facts	94%	86%
	False positive	Hallucinated reasoning trace or facts	6%	14%
Failure	Reasoning error	Wrong reasoning trace (including failing to recover from repetitive steps)	47%	16%
	Search result error	Search return empty or does not contain useful information	23%	-
	Hallucination	Hallucinated reasoning trace or facts	0%	56%
	Label ambiguity	Right prediction but did not match the label precisely	29%	28%

Knowledge-Intensive Reasoning Task: Result

1. ReAct > Act
2. ReAct vs CoT

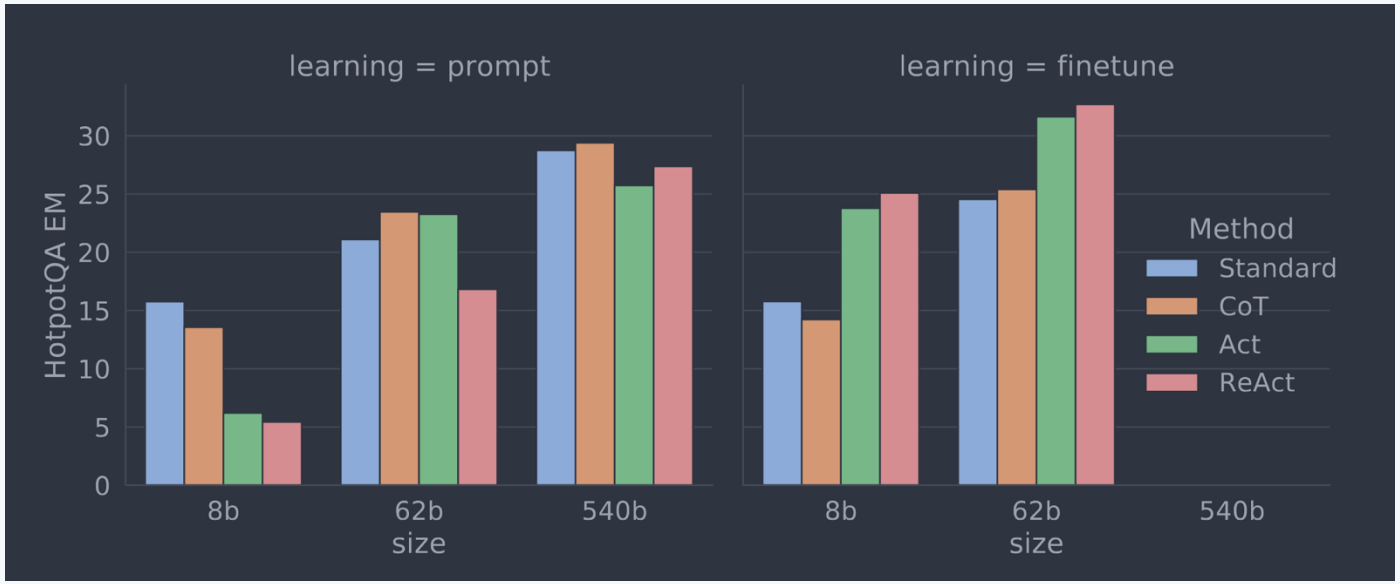
Internal and External Knowledge를 combine
← CoT가 hallucination에 취약하다는 점을 빼면, reasoning structure를 formulate하는 데에는 더 정확하기 때문

Prompt Method[a]	HotpotQA (EM)	Fever (Acc)
Standard	28.7	57.1
CoT (Wei et al., 2022)	29.4	56.3
CoT-SC (Wang et al., 2022a)	33.4	60.4
Act	25.7	58.9
ReAct	27.4	60.9
CoT-SC → ReAct	34.2	**64.6**
ReAct → CoT-SC	**35.1**	62.0
Supervised SoTA[b]	67.5	89.5



Knowledge-Intensive Reasoning Task: Result

1. ReAct > Act
2. ReAct vs CoT



Decision Making Task

1. ReAct > Act

Method	Pick	Clean	Heat	Cool	Look	Pick 2	All
Act (best of 6)	88	42	74	67	72	**41**	45
ReAct (avg)	65	39	83	76	55	24	57
ReAct (best of 6)	**92**	58	**96**	86	**78**	**41**	**71**
ReAct-IM (avg)	55	59	60	55	23	24	48
ReAct-IM (best of 6)	62	**68**	87	57	39	33	53
BUTLER$_g$ (best of 8)	33	26	70	76	17	12	22
BUTLER (best of 8)	46	39	74	**100**	22	24	37

ALFWorld

Method	Score	SR
Act	62.3	30.1
ReAct	**66.6**	**40.0**
IL	59.9	29.1
IL+RL	62.4	28.7
Human Expert	82.1	59.6

Webshop

Advantage

1. **Interpretability**: 모델의 사고 과정을 자연어로 확인 가능하여 왜 그런 행동을 했는지 진단하기 용이
2. **Trustworthiness**: 외부 지식에 기반한 추론으로 hallucination 억제 가능
3. **Controllability**: 인간이 중간 추론 과정을 수정하여 agent의 행동을 교정할 수 있음

Conclusion & Novelty

- ReAct는 LLM이 단순한 text generator를 넘어 external environment와 능동적으로 interact하여 문제를 해결하는 “지능형 agent”로 진화하는데 핵심 paradigm을 제시했다는 데에 의의를 가짐

향후 발전 방향

1. Fine-tuning
2. RL integration
3. Multi-modal
4. …