



MCP와 스킬을 사용한 OpenAlex 데이터 분석

에이전트와 함께 분석하는 학술 데이터: 직접 만든 OpenAlex MCP 서버와 스킬

박현우

서울대학교 데이터사이언스대학원

hyunwoopark@snu.ac.kr · <https://hyunwoopark.com>

제2회 OpenAlex Korean User Forum · 2026.06.23



발표자료 PDF



누적 토큰

22.6B

2026-03~06 합계 · ccusage

API 정가 환산

\$20,452

구독(Max) 사용 — 실지출 아님

월 최대

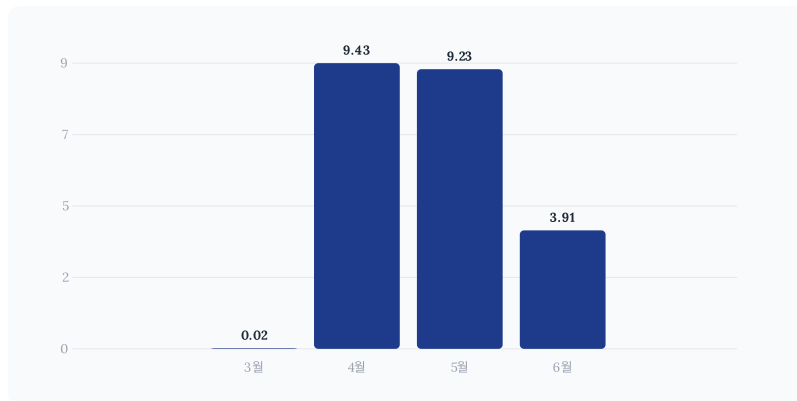
9.4B

2026-04 · 토큰/월

사용 모델

7

Claude-GPT + Open Weight Models



월간 토큰(10B 단위). 2026-04 일상 업무에 전면 도입 — 그 전엔 사실상 0.



누적 시청

87,222

2008-2026 · 19년치

최다 연도

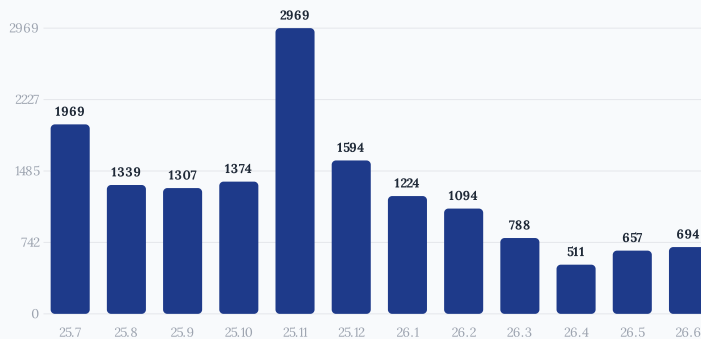
18,462

바로 작년(2025) — 역대 최다

2026 월평균

~830회

1-5월 · 2025의 절반 ▼



최근 1년 · 월별. 2025년 정점 → 2026년 봄 AI에 빠져 저점(4월 511회) → 요즘 재할 중. 6월은 보름 만에 694회(일 46건 — 5월의 2배 페이스).



01 문제 — 데이터가 아니라 '닿는 길'이 병목

OpenAlex는 충분히 큰데, 질문과 답 사이에 손이 너무 많이 간다

02 MCP — OpenAlex를 에이전트에게

제가 직접 만들어 npm에 올린 openalex-research-mcp 이야기

03 전환 — 업계 표준이 MCP에서 스킬로

무거운 상시 연결에서 가벼운 주문형 능력으로

04 스킬 — 같은 능력, 더 가볍게

마크다운 안내문 + 짧은 CLI 하나

05 실전 — 직접 돌려본 6가지 분석

문헌 리뷰·인용·저자·트렌드·기관·국가, 전부 라이브 데이터

PART 1

문제

데이터가 아니라 '데이터에 닿는 길'이 병목이다



수록 학술 성과물

240M+

매일 5만+ 건 추가

저자

90M+

기관·저널·인용까지 개방

데이터 비용

무료

오픈 데이터 · 키 무료

KISTI 성과물

7,902

피인용 786,341 (2026-06)

데이터는 이미 다 열려 있습니다(openalex.org). 그런데도 '내 질문'에 답하려면 매번 손이 많이 갑니다. 문제는 데이터가 아니라 데이터에 닿는 길입니다.

KISTI도 이 카탈로그 안에 있습니다 — 성과물 7,902건, 피인용 78만 회. 우리 데이터를 우리가 가장 빨리 분석할 수 있어야 합니다.



Works (W)

논문·성과물 — 제목·연도·인용·DOI



Authors (A)

저자 — 소속·ORCID·지표



Sources (S)

저널·리포지토리



Institutions (I)

기관 — ROR 연결



Topics (T)

주제 — 옛 concepts 대체



Publishers-Funders

출판사 · 연구비 지원기관

REST API · api.openalex.org

- `filter=` 조건 · `select=` 필드 · `group_by=` 집계
- 엔티티마다 고정 ID(W/A/S/I/T) — openalex.org 검색이나 CLI로 찾을 수 있음

초록 = `inverted-index`

- 전문(全文) 대신 {단어: [위치]} 형태로 제공(저작권)
- 위치대로 단어를 깔아 문장 복원 → 검색·요약엔 충분

무료·키 없이도 호출(이메일만 권장). 문서 docs.openalex.org · API api.openalex.org



어느 길이란 '질문'과 '답' 사이에 사람 손이 많이 들어갑니다.

1



웹 UI 클릭

탐색엔 좋지만 재현·대량 처리·자동화가 안 된다. 결과를 다시 쓰려면 또 클릭.

2



원시 REST API

filter · select · group_by 문법, 페이지네이션, inverted-index 초록 복원을 직접 다뤄야 한다.

3



스크립트 작성

한 번 쓰고 마는 파이썬 — 매번 새로 짜고, 유지보수가 쌓인다.

에이전트(MCP·스킬)는 이 셋 사이의 '사람 손'을 걷어냅니다 — 질문은 자연어로, 문법·복원·정리는 도구가.

PART 2

MCP

OpenAlex를 에이전트에게 — 자연어로 묻고 답 받기



Model Context Protocol: 에이전트(Claude 등)가 외부 도구·데이터에 접근하는 개방형 표준. USB-C처럼, 한 번 맞춰두면 어떤 클라이언트든 같은 방식으로 꽂힙니다.



자연어로 호출

사용자는 API 문법을 몰라도 된다. '이 주제 핵심 논문 찾아줘'면 끝.



스스로 조립

에이전트가 필터·정렬·후속 질의를 알아서 엮는다.



여러 클라이언트가 재사용

한 번 만들면 Claude Desktop·TypingMind 등에서 그대로.

핵심 그림: 사람은 '무엇을' 묻고, 에이전트가 '어떻게'를 OpenAlex API로 번역한다.



도구

31개

7개 범주

배포

npm

npx openalex-research-mcp

응답 최적화

~1.7KB

목록은 요약, 단건은 전체



문헌 검색·발견

불리언·저널·기관·인용 임계 필터



신뢰도 게이트

UTD24·FT50·AJG·Top AI 프리셋



인용 분석

전·후방 인용, 인용 네트워크



저자·기관

h-index·프로필·협업 네트워크



고가치 인용

리뷰·고전 논문, OA 버전



연구 지형·트렌드

주제 추이·분야 비교·지리 분포



엔티티 조회

작품·저자·기관·저널 단건 상세



프리셋 내장

저널·기관 그룹 큐레이션



claude_desktop_config.json

```
{
  "mcpServers": {
    "openalex": {
      "command": "npx",
      "args": ["-y", "openalex-research-mcp"],
      "env": { "OPENALEX_EMAIL": "you@example.com" }
    }
  }
}
```

✓ 사용자: "AI 안전 분야에서 2020년 이후 가장 영향력 있는 논문들 찾아줘" ✓ 에이전트: `get_top_cited_works(query=..., from_year=2020, min_citations=200)` 자동 호출 → 표로 정리



Claude Desktop·TypingMind 같은 클라이언트에선 **코드도 API 문법도 없이** 대화로 분석이 끝납니다.



자연어로 즉시 표

'AI 안전, 2020년 이후 인용 200+ 논문' 한마디 → 에이전트가 필터 걸어 표로



대화로 좁히기

'그중 오픈액세스만' · '우리 기관 저자만' → 후속 질문 한 줄로 정제



네트워크·인용 추적

'이 저자 협업 네트워크' · '이 논문 인용한 최근 리뷰'



기관·국가 비교

'A vs B 기관 생산성' · 'AI 논문 국가별 분포'

비개발자·탐색형 사용자에게 MCP가 가장 강한 지점 — **GUI에서 채팅으로 끝.** (자동화·파이프라인이 필요하면 다음 장부터의 스킬+CLI.)

PART 3

전환

그사이, 업계 표준이 MCP에서 '스킬'로 옮겨갔다



MCP의 한계

- 도구 정의 30여 개가 **항상 컨텍스트에 상주** → 토큰 비용 고정 지출
- 전용 서버 프로세스 + 클라이언트 설정 → **설치·결합 장벽**
- GUI 클라이언트(데스크톱)에 묶이는 경향

스킬의 이점

- **필요할 때만 로드**(progressive disclosure) → 안 쓰면 토큰 0
- 셸/CLI와 결합 → 어디서나, 자동화·파이프라인에 그대로
- 마크다운 + 작은 스크립트면 끝 → **만들고 고치기 쉽다**

01 2024 — MCP

에이전트-도구 연결의 공통 규격이 자리잡다

02 2025 — 폭증과 피로

서버는 늘었지만 상시 로드 비용·설정 복잡도가 드러나다

03 2026 — 스킬

필요할 때만 펼치는 스킬 + CLI가 사실상의 기본값

핵심은 'MCP가 틀렸다'가 아니라, **무거운 상시 연결에서 가벼운 주문형 능력**으로 무게중심이 이동했다는 것.

PART 4

스킬

같은 OpenAlex 능력을, 훨씬 가볍게



놓고 있을 땐 토큰 0

호출하는 순간에만 안내문이 펼쳐진다. 평소엔 비용이 들지 않는다.



귀찮은 건 CLI가 대신

API 키 주입 · select= 필드 트리밍 · 초록 복원 · DOI→ID 해석을 자동으로.



어디서나 물린다

터미널이 있으면 크론·파이프·다른 에이전트에도 그대로.



같은 기능, 다른 포장

MCP 서버가 하던 일을 '스킬 + 작은 명령어'로 옮긴 셈.

그럼 이 스킬과 그 도구(CLI)가 실제로 어떻게 생겼는지, 그리고 6가지 데모를 차례로 보시죠.



```
---
name: openalex
description: "OpenAlex로 논문·저자·기관·주제 검색.
  논문 검색·저자 찾기·인용 확인·피인용 수·학술 트렌드..." # 한국어 트리거
allowed-tools: Bash(openalex *), Bash(curl *), Bash(jq *)
---

## When NOT to use      # 전문은 jina, BibTeX는 CrossRef ...
## Core Patterns — openalex CLI
##   works · authors · institutions · topics · cites · trend · abstract
##   (키 주입 · select= 트리밍 · 초록 복원 · DOI→ID 자동)
```

스킬 = 마크다운 안내문 한 장. 호출 순간에만 로드 → 평소 토큰 0. description의 한국어 트리거가 자동 활성화를 좌우합니다.



공식 CLI는 없습니다. OpenAlex는 REST API만 제공 → 그 위에 얇은 래퍼를 직접 만들었습니다(`~/dotfiles/bin/openalex` , 파이썬 한 파일).



works · work

논문 검색 · 단건 조회



authors

저자 검색 → A-id



institutions·source·topics

기관·저널·주제 → ID



cites · trend · abstract

인용 · 연도추이 · 초록 복원

CLI가 대신 처리: API 키 주입 · `select=` 필드 트리밍(응답 3× 축소) · 초록 복원 · DOI→ID. 이제 보실 6개 데모는 전부 이 한 줄입니다.

PART 5

실전 — 직접 돌려본 6가지

질문 하나로 이런 게 금방 나옵니다 (전부 실제 출력)



질문 — "LLM 분야에서 가장 많이 인용된 논문은?"

```
$ openalex works "large language models" \  
  --filter "publication_year:>2021,cited_by_count:>2000"  
(total: 56)  
4,977 ChatGPT for good? ... LLMs for education 2023  
3,274 Large language models in medicine · Nat Medicine 2023  
3,134 LLMs encode clinical knowledge · Nature 2023  
2,435 A Survey on Evaluation of LLMs 2024
```

주제어 + 연도 + 인용 임계 한 줄 → 분야 핵심 논문이 인용순으로 정렬. 검색부터 정리까지 30초.



질문 – "이 OpenAlex 소개 논문을 누가 인용했나?" (총 278편 중 계량서지·OA 사례)

```
$ openalex cites 10.48550/arXiv.2205.01833 # OpenAlex 소개 논문
(total: 278)
530 A meta systematic review of AI in higher education 2024
 58 Who are the 100 largest scientific publishers? · J.Doc 2022
 57 The APC-barrier & stratification in OA publishing · QSS 2023
 56 The landscape of biomedical research · Patterns 2024
```

DOI 하나면 전방 인용 278편이 바로 – 계량서지·오픈액세스 연구들이 OpenAlex를 토대로 쌓이는 흐름이 보입니다.



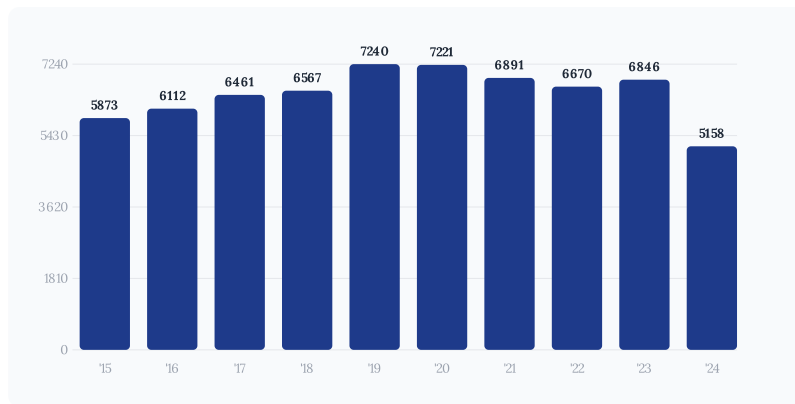
이름은 모호해서(동명이인 27명) `openalex authors "Ludo Waltman"` 로 검색 → 393편짜리가 그분 → ID `A5027467000` (피인용 45,746). 대표작이 곧 이 청중께 익숙한 도구들입니다.

대표 논문	연도	피인용
VOSviewer — 계량서지 매핑 소프트웨어	2009	19,703
Visualizing Bibliometric Networks	2014	3,203
The Leiden Manifesto for research metrics	2015	2,626
CitNetExplorer/VOSviewer 클러스터링	2017	2,564

검색 → ID 고정 → 대표작·피인용까지 한 줄 흐름. 이름은 흔들려도 ID는 안 흔들립니다.



주제 ID는 `openalex topics "bibliometrics"` → **T10102** (scientometrics & bibliometrics research).



`openalex trend T10102` 한 줄로 연도별 추이. 단, 최근 연도(2024~)는 색인 지연으로 과소집계 — 해석 시 주의.



기관	성과물	총 피인용
서울대학교 (I139264467)	241,746	20,952,350
KISTI (I878022262)	7,902	786,341
서울대병원 (I2802835388)	47,127	3,664,964

기관 ID는 `openalex institutions "서울대학교"` 한 줄로 — 검색 결과 URL 끝이 곧 ID. 성과물·피인용도 즉시 나옵니다.



`group_by=authorships.countries` 한 번으로 국가별 분포. 연구 지형·정책 비교에 바로 쓰는 그림.



유의할 점

- 데이터 **커버리지·품질 편차** — 분야·언어별로 다르다
- 초록은 inverted-index({단어:위치})에서 복원 — **원문 전문(全文)**은 아니다 (앞 OpenAlex 개요 참고)
- 최근 연도 **색인 지연** · 숫자는 반드시 **원본 검증**(환각 가드)

2026 동향

- OurResearch → **OpenAlex 중심 리브랜딩** (Unpaywall 등 통합)
- **Data Version 2** 및 신규 코드베이스 'Walden'
- MCP·스킬 등 **에이전트 접근 경로**가 빠르게 표준화 중

필터 = 연도·인용수·기관·OA 여부 같은 조건(예: `cited_by_count:>200`). **프리셋** = 검증된 저널·기관 묶음(UTD24·FT50·AJG·Top-AI). 도구가 빨라질수록 이 신뢰도 게이트 + 원본 확인이 더 중요합니다.



상황	MCP가 낫다	스킬 + CLI가 낫다
사용 환경	Claude Desktop 등 GUI 클라이언트	터미널·서버·자동화 파이프라인
사용자	비개발자, 대화형 탐색 위주	개발자, 재현·배치 처리 위주
토큰 비용	상시 로드(고정 지출)	주문형(놀 땐 0)
조합·확장	클라이언트 안에서	셀·크론·다른 에이전트와 자유롭게

정답은 택일이 아니다 — **같은 능력을 두 포장으로** 두고, 상황에 맞게 꺼내 쓰면 된다.



1



진입장벽이 급감한다

OpenAlex + 에이전트 = 코드가 아니라 '질문'으로 일한다. 6개 데모가 전부 한 줄.

2



패턴이 핵심이다

MCP냐 스킴이냐보다, 신뢰도 게이트·검증·재현이 데이터 품질을 지킨다.

3



오늘 시작할 수 있다

무료 데이터 + 공개 도구, 키 하나면 끝.
github.com/oksure/openalex-research-mcp.

에이전트라는 단어를 열 번 듣는 것보다, 한 줄 돌려 표가 나오는 걸 한 번 보는 게 빠릅니다.



감사합니다

질문 환영합니다 · Q&A



hyunwoopark.com



발표자료 PDF

박현우

서울대학교 데이터사이언스대학원

hyunwoopark@snu.ac.kr · <https://hyunwoopark.com>