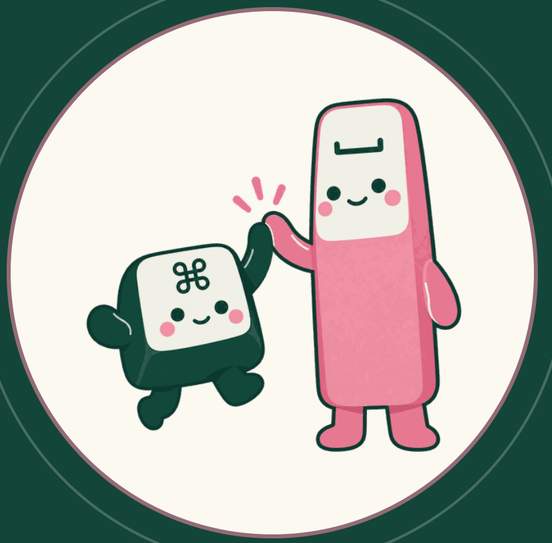
앞서가는 연구자의 지식 관리 시스템

구요한 교수의 연구 WIKI와 에이전트. 논문이 연구 아이디어와 보고서로 이어지는 구조를 실제 화면으로 봅니다.



01
1부
LLM Wiki로 논문을 컴파일하기

02
2부
메인 볼트에 이어 연구 질문 찾기

03
3부
연구 에이전트로 운영하기

논문을 모았는데
연구는 왜 그대로일까

모으고, 읽고, 관리하는 일은 서로 다른 일입니다. 셋을 한 폴더가 떠안으면 셋 다 멈춥니다.



01 **모으기** 다운로드 폴더에 PDF가 쌓인다

02 **읽기** 요약은 그때그때 만들고 사라진다

03 **쓰기** 필요할 때 다시 찾지 못한다

핵심 포인트

지난달 다시 꺼내 쓴 논문이 몇 편인지 세어 보십시오.

오늘의 지도 한 장

원문에서 산출물까지 네 칸입니다. 칸마다 맡는 일이 다르고, 오늘은 왼쪽에서 오른쪽으로 갑니다.

STEP 01
원문 서재
Zotero 안의 논문
PDF와 서지, 원본 그대로

STEP 02
LLM Wiki
같은 템플릿으로
컴파일한 문헌 지식

STEP 03
메인 볼트
내 프로젝트와 내 해석

STEP 04
연구 에이전트
수집, 점검, 브리핑을 운영

핵심 포인트

내 자료가 지금 네 칸 중 어디에 멈춰 있는지 표시해 보십시오.

역할 경계

원문: 고치지 않음 / **위키**: AI가 정리 / **볼트**: 내가 해석 / **에이전트**: 제안하고 사람이 확정

매일 쓰는 시스템을 그대로 엽니다

오늘 화면은 시연용으로 따로 만든 것이 아니라 제가 실제로 운영하는 볼트와 위키입니다.

01
메인 볼트
10,000개가 넘는 노트

02
LLM Wiki
1,100여 개 페이지

03
논문 분석
원문 인용을 붙인 16편

04
운영 기간
첫 데일리
노트부터 4년 넘게

핵심 포인트

오늘 보는 화면은 모두 실제로 쓰는 시스템에서 가져왔습니다.

1부 LLM Wiki 구축

논문과 기술자료를 같은 템플릿으로 컴파일합니다.

원문 서재는 Zotero가 지킵니다

논문 서지, PDF, 하이라이트는 Zotero 한 곳에 둡니다. 위키와 볼트는 인용 키 하나로 원문을 가리킵니다.

STEP 01
Zotero
논문 서지, PDF,
하이라이트를 원본으로 보관

STEP 02
인용 키
Better BibTeX가 논문마다
바뀌지 않는 키를 붙임

STEP 03
플러그인
CMDS Zotero로 인용,
주석 가져오기, PDF 쪽 열기

STEP 04
위키와 볼트
본문에는 인용 키만,
서지는 Zotero에서 읽음

핵심 포인트

논문 서지는 Zotero 한 곳에만 두고, 노트에는 인용 키만 남기십시오.

역할 경계

Zotero: 서지 원본, 플러그인은 읽기만 / **위키와 볼트**: 인용 키로 원문을 가리킴

요약은 압축이고
위키는 컴파일입니다

요약은 한 번 쓰고 사라지는 결과물입니다. 위키는 원문을 그대로 두고 규칙대로 쪼개 쌓는 지식입니다.



01 **요약** 대화 안에서 만들고 대화와 함께 묻힘

02 **검색** 질문할 때마다 원문을 다시 뒤짐

03 **위키** 미리 정리해 두고 계속 덧붙임

핵심 포인트

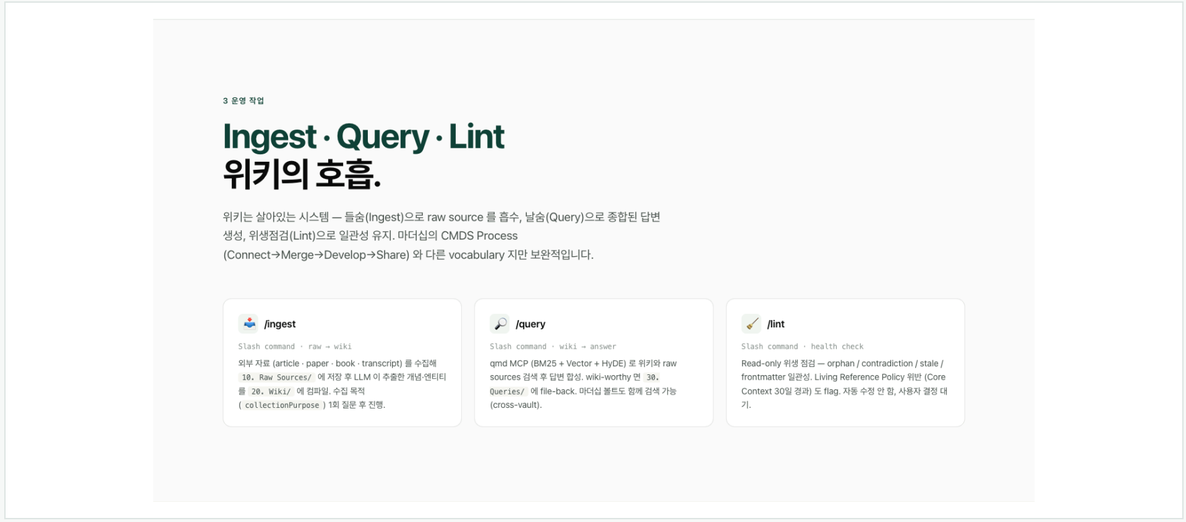
다음 질문이 지난 정리 위에서 시작되게 만드십시오.

역할 경계

원문: 바꾸지 않는다 / **AI**: 규칙대로 쪼개고 잇는다 / **연구자**: 규칙을 정한다

LLM Wiki의 세 층

원문, 위키, 규칙. 위키의 품질은 모델보다 규칙 파일이 정합니다.



LLM Wiki 공개 안내 사이트의 운영 작업 설명 (2026-09-10 캡처)

01 원문 논문과 자료를 바꾸지 않고 보관

02 위키 개념, 주장, 방법을 페이지로 연결

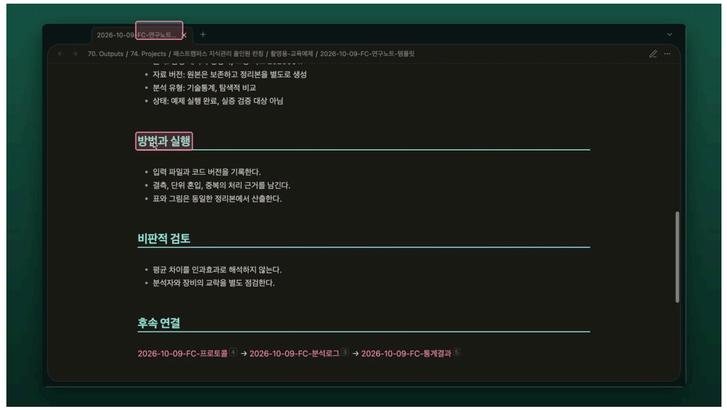
03 규칙 무엇을 어떻게 남길지 적은 파일

핵심 포인트

위키를 만들기 전에 남길 항목부터 적어 두십시오.

논문 한 편을 같은 항목으로 남깁니다

템플릿이 곧 연구 기준입니다. 같은 칸이 있어야 나중에 여러 편을 나란히 놓을 수 있습니다.



연구 노트 템플릿 (강의 촬영용 교육 예제, 합성 데이터)

01 질문과 방법 무엇을 어떻게 물었나

02 주장과 근거 주장마다 원문 위치

03 한계와 관계 지지, 반박, 불확실

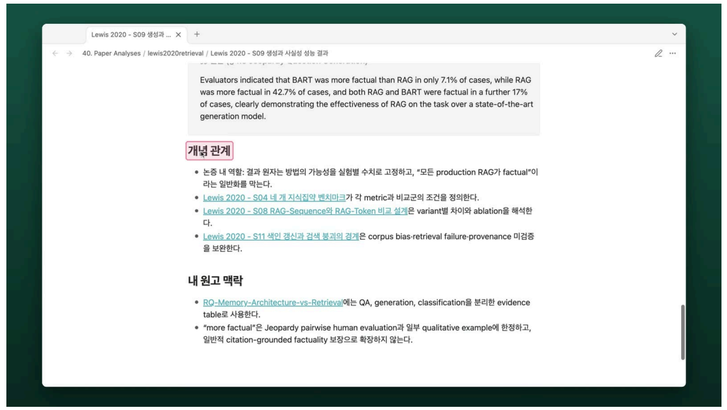
04 내 연구 어디에 쓸 수 있나

핵심 포인트

내 분야에 맞는 항목 다섯 개를 오늘 정해 보십시오.

주장마다
원문 위치를 남깁니다

요약문보다 근거의 위치가 오래 갑니다. 나중에 인용할 때 다시 원문을 뒤지지 않아도 됩니다.



Lewis 2020 RAG 논문 결과 노트 (위키 논문 분석, 검증 전 표기)

01 원문 인용 주장 옆에 문장 그대로

02 개념 관계 다른 논문과 무엇이 이어지나

03 내 원고 맥락 어디에 인용할 수 있나

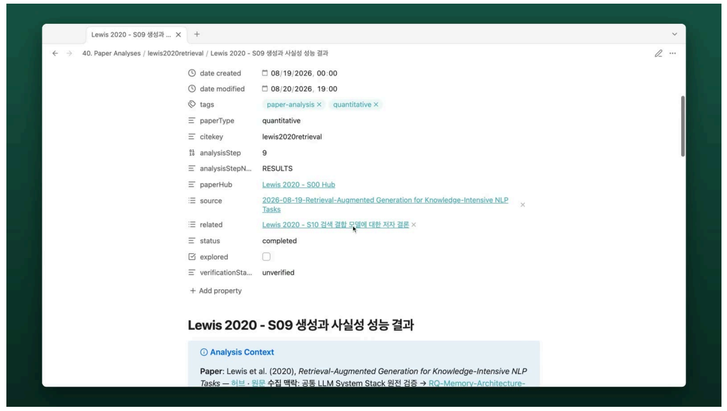
핵심 포인트

요약 한 줄보다 원문 위치 하나를 먼저 남기십시오.

실제 위키 화면 정지 컷 / 검증 전 표기는 그대로 둡니다.

한 편을 허브와 단계 노트로 나눕니다

논문 한 편을 긴 요약 하나로 두지 않고, 허브 한 장과 단계별 노트로 나눠 필요한 단계만 꺼내 씁니다.



논문 단계 노트의 속성. 허브, 원문, 관련 노트가 링크로 이어진다

01 허브 논문 전체의 입구

02 단계 노트 방법, 결과, 한계를 따로

03 속성 원문, 허브, 검증 상태를 기록

핵심 포인트

긴 요약 하나보다 꺼내 쓸 수 있는 조각 여러 개가 낫습니다.

실제 위키 화면 정지 컷 / 검증 전 표기는 그대로 둡니다.

오늘 밤 시작점
LLM Wiki 스타터킷

오늘 본 세 층 구조를 빈 볼트로 공개해 두었습니다. 내려받아 Obsidian으로 열면 바로 시작할 수 있습니다.

휴대폰 카메라로 찍으면 스타터킷 페이지가 열립니다.

01 **구조** 원문, 위키, 질의, 논문 분석 폴더

02 **규칙** 에이전트가 읽는 규칙 파일

03 **명령** 수집, 질의, 점검 절차

핵심 포인트

오늘은 논문 한 편만 넣고 규칙대로 정리되는지 보십시오.

2부 메인 볼트 연결

위키의 문헌 지식을 내 프로젝트와 관심 주제에 잇습니다.

위키는 넓게,
볼트는 깊게

위키는 AI가 넓게 정리한 참고 층이고, 메인 볼트는 내가 쓴 해석의 층입니다. 섞지 않고 잇습니다.



01 **참고 층** 위키, 많은 논문을 넓게 정리

02 **해석 층** 메인 볼트, 내 문장과 프로젝트

03 **연결** 둘을 오가는 링크와 질문

핵심 포인트

AI가 쓴 정리와 내가 쓴 해석을 한 폴더에 섞지 마십시오.

역할 경계

위키: AI가 쓰고 사람이 검토 / **메인 볼트**: 사람이 쓰고 AI가 돕는다

두 층을 잇는 세 가지 장치

링크 하나로 끝나지 않습니다. 질문, 양방향 링크, 교차 검색이 함께 있어야 연결이 살아 있습니다.

01
연구 질문 페이지
근거와 프로젝트를 묶는 매듭

02
양방향 링크
볼트에서 위키로, 위키에서 볼트로

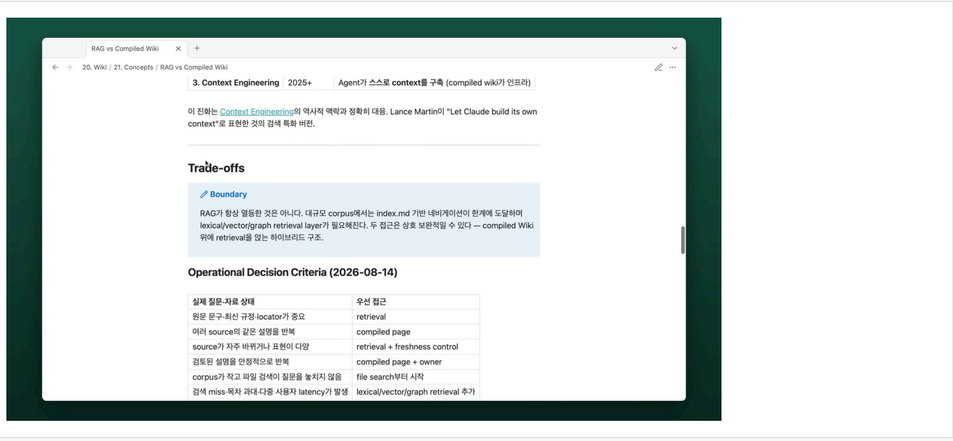
03
교차 검색
질문 하나로 두 볼트를 함께

핵심 포인트

연구 질문 하나를 페이지로 만들고 관련 논문을 거기에 모으십시오.

질문 하나가
두 층을 묶습니다

"언제 검색이 필요하고 언제 위키로 충분한가"라는 질문이 개념 페이지와 판단 기준 표로 자랐습니다.



위키 개념 페이지 'RAG vs Compiled Wiki'의 경계와 판단 기준

01 질문 위키가 커지면 검색이 필요한가

02 근거 여러 원문의 주장을 한 페이지에

03 기준 상황별 판단 표로 정리

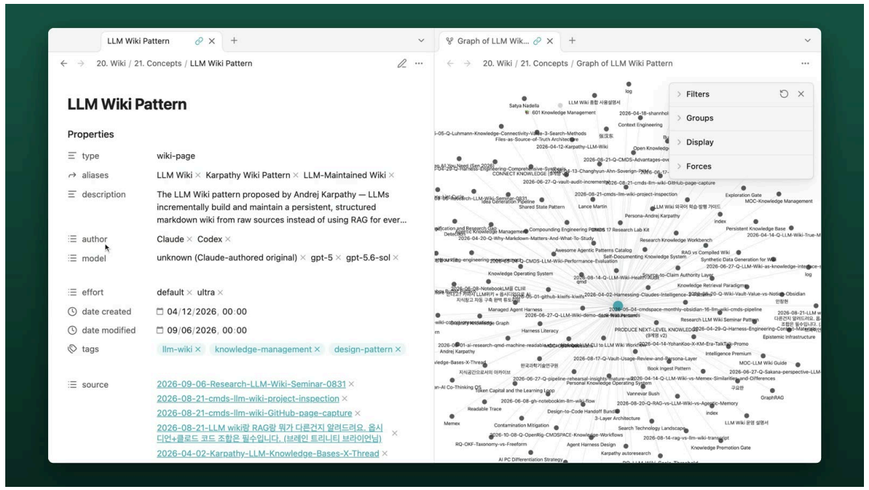
핵심 포인트

질문을 문장으로 적으면 근거가 그 아래로 모이기 시작합니다.

실제 위키 화면 정지 컷 / 검증 전 표기는 그대로 둡니다.

그래프를 따라 이어서 연구합니다

한 페이지에서 출발해 링크로 이어진 주변 개념을 따라가면 다음에 읽을 것이 보입니다.



LLM Wiki Pattern 페이지와 로컬 그래프 (실제 위키)

01 출발 지금 읽는 개념 페이지

02 주변 링크로 이어진 개념과 원문

03 다음 아직 얇은 연결이 다음 읽을거리

핵심 포인트

다음에 읽을 논문은 그래프의 얇은 가장자리에서 고르십시오.

실제 위키 화면 정지 컷 / 검증 전 표기는 그대로 둡니다.

라이브: 질문 하나로 두 볼트를 찾습니다

연구 질문 하나를 에이전트에게 주고 위키 근거와 메인 볼트 노트를 함께 찾게 합니다.

질문: 위키가 커지면 언제부터 검색이 필요해지는가

어디서	찾은 노트
위키 (AI가 정리)	Wiki Search Methods Comparison
위키 (AI가 정리)	RQ: LLM Wiki 패턴이 수백 소스·수천 페이지에서도 index.md 네비게이션만으로 충분한가
위키 (AI가 정리)	LLM Wiki 운영 설명서
위키 (AI가 정리)	Ingest-Query-Lint Cycle
메인 볼트 (내가 씀)	**검색은 휘발성, 위키는 자산**
위키 (AI가 정리)	LLM Wiki Pattern
위키 (AI가 정리)	Obsidian Tooling for LLM Wiki
위키 (AI가 정리)	MOC-LLM Wiki Knowledge Interface Lecture

두 볼트를 한 번에 검색한 사전 실행 결과 (2026-10-15, qmd 교차 검색, 상위 8건)

같은 질문의 사전 실행 결과 (2026-10-15). 라이브가 막히면 이 화면으로 설명

01 질문 미리 정한 연구 질문 하나

02 검색 위키와 메인 볼트를 동시에

03 결과 근거 페이지와 내 노트가 함께

핵심 포인트

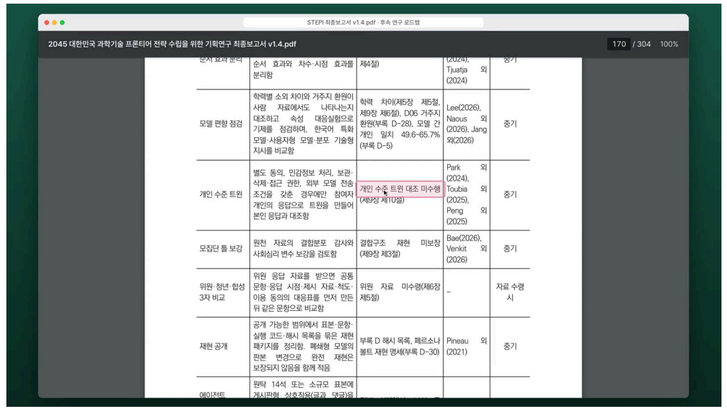
검색 결과에서 위키 근거와 내 노트가 나란히 나오는지 보십시오.

역할 경계

에이전트: 두 볼트를 찾고 묶는다 / **연구자**: 무엇을 쓸지 고른다

같은 칸에 넣으면 빈칸이 보입니다

선행 연구 8편과 이번 연구를 같은 7개 칸에 넣자 확인하지 못한 칸이 드러났고, 그 칸이 후속 연구 과제가 됐습니다.



STEPI 기획연구 최종보고서의 후속 연구 로드맵 표 (커맨드스페이스 수행)

01 같은 칸 선행 연구와 본 연구를 같은 열로

02 미확인 없다가 아니라 이번에 확인 못 함

03 후속 과제 빈칸이 로드맵의 항목으로

핵심 포인트

빈칸은 공백 확정이 아니라 확인할 후보로 다루십시오.

STEPI 기획연구 실제 작업 화면 / 커맨드스페이스 수행, 판단은 연구진 초안

빈칸에서 후속 연구까지

비교표의 빈칸을 주장과 근거의 지도로 확인하고, 실행할 수 있는 연구 계획으로 바꿉니다.

STEP 01
비교표
같은 항목으로 문헌을 나란히

STEP 02
근거 지도
지지, 반박, 불확실로 분류

STEP 03
공백 후보
근거가 얇거나 엇갈린 자리

STEP 04
후속 설계
질문, 방법, 필요한 자료

핵심 포인트

지금 붙잡은 질문 하나를 채팅에 한 줄로 적어 보십시오.

3부 연구 에이전트

문헌 조사부터 보고서 초안까지, 연구 기록을 운영하는 에이전트를 봅니다.

제 에이전트는 하네스 위에서 일합니다

에이전트는 모델에 하네스(실행 루프, 도구, 맥락, 권한)를 더한 실행 단위입니다. 스킬은 에이전트가 쓰는 절차입니다.

01
하네스
Claude Code와 Codex, 맥 두 대에 작업 리그 8종

02
에이전트
9요한 전문 에이전트 9개와 지휘 역할 하나

03
스킬
덱 제작, 논문 반입, 수신함 처리 등 120여 개

04
대표작
기획연구 지원, 모델 블라인드 판정, 매일 브리핑

핵심 포인트

에이전트를 만들기 전에 반복되는 절차부터 스킬로 적어 두십시오.

역할 경계

하네스: 실행 환경과 권한 / **에이전트**: 계획하고 실행 / **스킬**: 절차 / **사람**: 승인

본인 프레임워크와 실제 운영 시스템 기반 설명입니다.

Anthropic: 에이전트 하네스 설계 Claude Code 스킬 문서 9요한 에이전트 설계 에이전트와 하네스 소개

업무마다
스킬 원장이 있습니다

스킬 121개를 업무별 원장에 적어 둡니다. 볼트 규칙을 지키는 기반 스킬 18개는 네 영역이 함께 씁니다.

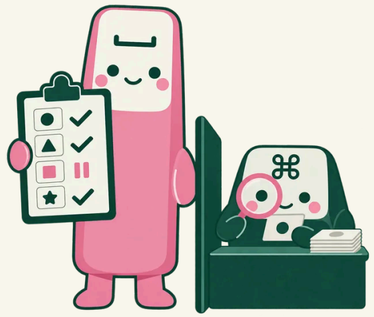
01	**연구 20개**	multi-research, research-verify
02	**강의와 글 30개**	cmds-lecture-builder, thebetter-writer
03	**프로젝트 31개**	cmdspace-web-builder, ask-inbox
04	**일반업무 22개**	cmds-email, meeting-minutes

핵심 포인트

반복하는 업무 하나를 골라 첫 스킬로 적어 보십시오.

세 에이전트와 한 가지 약속

에이전트는 연구를 대신하지 않고 연구 기록을 운영합니다. 바꾸는 일은 사람의 승인 뒤에만 합니다.



01 **수집·정리** 새 자료를 모아 위키에 반영

02 **유지보수** 오류, 누락, 바뀐 근거를 점검

03 **브리핑** 현재 상태와 다음 행동을 보고

핵심 포인트

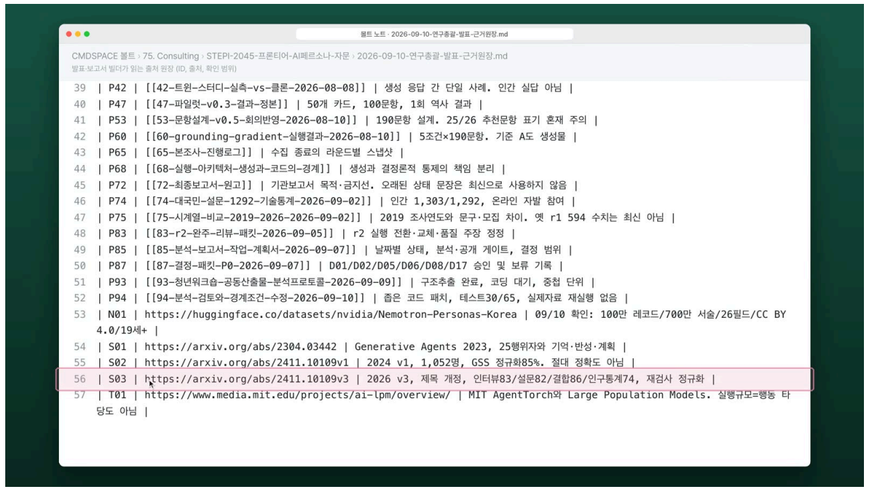
에이전트에게 읽기는 맡기고 쓰기는 승인 뒤에만 허락하십시오.

역할 경계

에이전트: 찾고, 점검하고, 제안한다 / **연구자**: 확정하고 승인한다

숫자 하나에서 원문 초록까지

보고서에 쓴 숫자가 어느 원문의 어느 문장에서 왔는지 근거 원장으로 세 번 만에 따라갑니다.



STEPI 기획연구 근거 원장. 보고서 수치마다 원문 주소, 확인 날짜, 범위를 기록

01 보고서 문장 숫자와 출처 표기

02 참고문헌 원문 주소로 이동

03 근거 원장 확인한 날짜와 범위

핵심 포인트

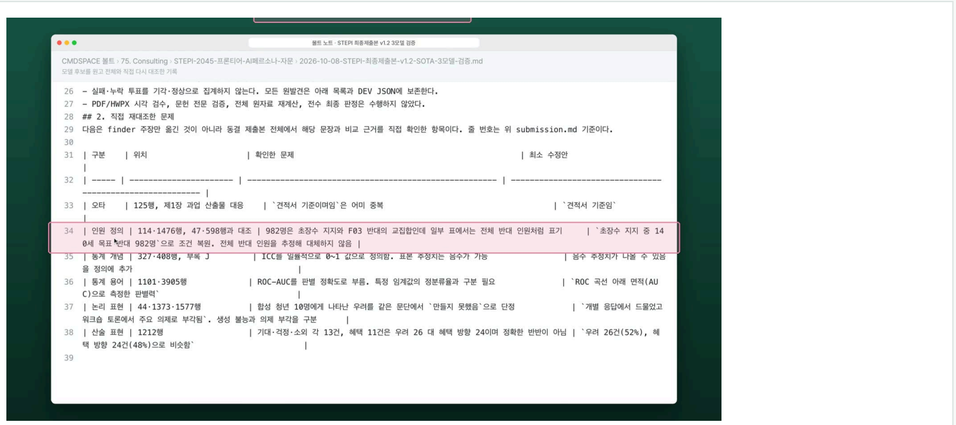
보고서의 숫자마다 원문으로 돌아가는 길을 남기십시오.

STEPI 기획연구 실제 작업 화면 / 커맨드스페이스 수행, 판단은 연구진 초안

Park et al. 2024, arXiv STEPI 2045 공개 아카이브

에이전트가 먼저 짚고 사람이 확정합니다

제출 전 원고를 여러 모델이 따로 읽고 문제 후보를 냈습니다. 후보는 오류가 아니고, 확정은 사람이 원고와 대조해서 했습니다.



STEPI 제출 전 원고 검증 노트. 모델이 짚은 후보를 사람이 원고와 다시 대조한 표

01 후보 여러 모델이 같은 문장을 따로 짚음

02 구분 후보 수는 오류 수가 아님

03 확정 사람이 원고 전체와 재대조

핵심 포인트

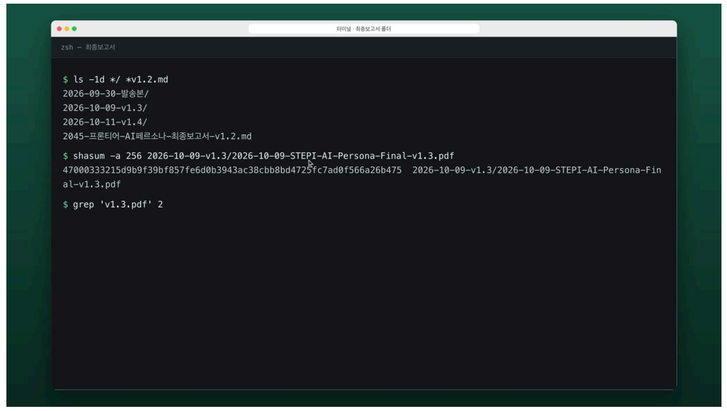
모델이 짚은 것은 후보로, 확정은 원문 대조 뒤에 하십시오.

역할 경계

모델: 문제 후보를 낸다 / **연구자**: 원고와 대조해 확정한다

받아들이고, 따르지 않고, 원본은 남깁니다

지적은 받아들이되 고치는 방식은 사람이 정하고, 고치기 전 판은 지우지 않고 지문으로 보존합니다.



STEPI 보고서 판 폴더와 SHA-256 지문 확인 (터미널 실제 실행)

01 받아들임 맞는 지적은 다음 판에 반영

02 따르지 않음 숫자는 두고 빠진 단계를 보탬

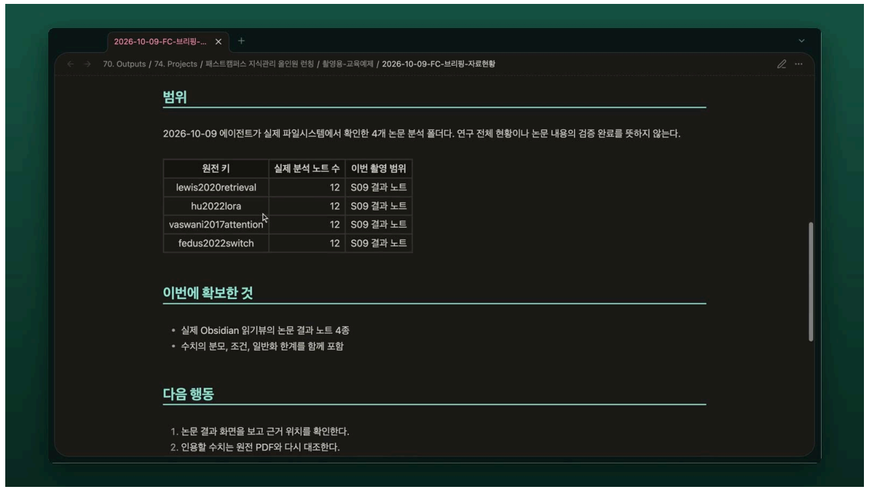
03 보존 이전 판은 지문으로 확인

핵심 포인트

고칠 때마다 이전 판을 남겨 언제든 돌아갈 수 있게 하십시오.

매일 브리핑으로 다음 행동을 받습니다

에이전트가 실제 파일을 점검해 무엇이 있고 무엇이 비었는지, 다음에 무엇을 할지 보고합니다.



연구 현황 브리핑 노트 (강의 촬영용 예제, 에이전트가 실제 파일을 점검한 결과)

01 범위 무엇을 점검했는지 먼저

02 확보 실제로 있는 것만 적기

03 다음 행동 우선순위 순으로 제안

핵심 포인트

브리핑에는 점검 범위와 확인하지 못한 것을 꼭 적게 하십시오.

오늘 본 세 단계를 직접 만드는 길

오늘은 완성된 시스템을 보여 드렸습니다. 강의에서는 시작 볼트에서 출발해 같은 구조를 내 연구로 만듭니다.



01 **저장소** Obsidian 연구 저장소 설계

02 **연구 WIKI** AI가 이해하는 위키와 비교표

03 **에이전트** 수집, 유지보수, 브리핑 3종

04 **완성** 내 연구 질문으로 끝까지

핵심 포인트

내 연구 질문 하나로 네 칸을 끝까지 한 번 돌려 보십시오.

오늘 밤 열어 볼 것

강의 자료와 함께 오늘 이야기의 바탕이 된 공개 자료를 네 가지 남깁니다.

01
위키 스타터킷
세 층 구조의 빈 위키

02
시스템 파일
에이전트가 읽는
규칙 공개본

03
AKM Index
내 시스템 자가 점검

04
지산 블로그
오늘 논지의 에세이 원문

핵심 포인트

네 가지 중 하나만 오늘 밤 열어 보십시오.

질문을 받습니다

채팅에 남겨 주신 연구 질문과 오늘 내용에 대한 질문을 함께 받습니다.

01	**연구 질문**	지도 위에서 어디서 시작할지
02	**오늘 내용**	위키, 볼트, 에이전트
03	**환경**	비용, 보안, 운영체제

감사합니다

오늘 본 구조의 공개 자료와 다음 이야기는 아래 주소로 이어집니다.



강의 자료

labs.cmdspace.work/
fc-seminar-1015/

커맨드스페이스

cmdspace.work

강사 소개

bio.cmdspace.work