

- 생성형 AI 활용 논문 작성 방법 · 오피디언과 LLM WIKI로 연구 지식 베이스 만들기

AI에게 자료를 붙여넣지 말고, 자료가 쌓이는 그릇을 지어라.

흩어진 연구자료를 AI가 검색·연결하는 나만의 지식 베이스로 만들고, 그 위에서 논문 작성까지.
제안이 아니라 **3년 · 10,000+** 노트로 운영 중인 시스템입니다.

● 오늘 이 분과의 존재 이유

이용자 · HEPI 2025

92%

학생이 생성형 AI를 **사용**한다 (전년 66% → 2026년판 95%). 88%는 과제에 활용. 사용률이 부정행위율이 아니다

도서관 · ITHAKA S+R 2025

31%

AI 생성물 평가를 **가르칠 자신이 있다**는 응답. 미디어 조작 평가는 51%. 재정 부족이 최대 장벽 81%

이용자의 92%는 이미 쓰고 있는데, 그걸 가르칠 준비가 됐다는 도서관은 31%입니다. **이 격차를 메우는 것이** 오늘의 주제입니다.

● 오늘 보실 시스템은 이 네 가지 규칙 위에 서 있습니다

설계 원리 — 이 구조를 지탱하는 네 가지 규칙

원칙 1 · 불변

원본은 고치지 않는다

수집한 원자료는 편집 대상이 아니다.
해석은 별도 층에 쌓는다. 원본이 흔들리면 모든 인용의 근거가 무너진다.

원칙 2 · 분업

사람은 판단, AI는 정리

무엇을 넣고 무엇을 물을지는 사람이,
요약·분류·상호참조·색인은 AI가. 이
경계가 흐려지면 시스템이 아니라 위
임이 된다.

원칙 3 · 추적

모든 주장에 출처를

답변은 근거 링크와 함께 나온다. 어
느 원자료의 몇 페이지에서 왔는지
되짚을 수 없으면 학술적으로 쓸 수
없다.

원칙 4 · 누적

좋은 답은 다시 자산으로

질의 결과가 대화창에서 끝나지 않고
위키 페이지가 된다. 물을수록 커지는
구조 — 붙여넣기와의 결정적 차이다.

이 네 규칙은 **도구가 바뀌어도 남습니다**. 그래서 오늘은 도구 강의가 아니라 인프라 이야기입니다.

- 남의 도구 배우기가 아니라, 본업의 AI 확장입니다

사서는 이미 지식관리자다 — 본업이 곧 구조다

수집 · 장서 개발

→ Raw Sources

원자료 서고. 논문 PDF. 웹클리프.
메모를 불변으로 보존한다.

분류 · 메타데이터

→ Wiki 컴파일

AI가 개념·요약·태그로 정리된
주제 파일을 쓰고 유지한다.

연결 · 상호참조

→ 위키링크

자료 카드 사이의 상호참조. 사
람이 손으로 하던 연결을 AI가
잇는다.

검색 지원 · 참고봉사

→ Query

자연어 질문에 근거(출처)와 함
께 답한다. 참고봉사의 확장.

큐레이션

→ 사람의 판단

무엇을 넣고 무엇을 물을지 —
AI가 대신하지 않는 유일한
자리.

여러분이 평생 해오신 일의 구조 그대로입니다. 다른 건 **분류와 연결의 노동을 AI가 맡는다는 것**뿐입니다.

- 같은 질문을 세 번 던지겠습니다

"이 분야 선행연구 정리해줘."

같은 질문, 같은 모델입니다. 다른 것은 그 모델이 딴고 선 구조뿐입니다.

- 프롬프트에 머물면 자료가 쌓이지 않습니다

같은 질문, 세 개의 답 — "이 분야 선행연구 정리해줘"

V1 · 프롬프트

빈 채팅창에 질문만

일반론 답변. 대화창에 남았다 사라진다. 다음에 또 처음부터 — 자료가 쌓이지 않는다.

V2 · 컨텍스트

자료를 붙여넣고 질문

이번 한 번은 정확하다. 그러나 매번 손으로 맥락을 다시 넣어야 한다.

V3 · 하네스

지식 베이스에서 질문

한 줄만 던져도 근거(출처 링크)와 함께 답한다. 물을수록 지식이 쌓인다.

AI에 실망하는 이유는 모델이 나빠서가 아니라 **v1에 머물러서**입니다. 차이는 모델이 아니라 구조입니다.

- V3의 구조 · 카파시 LLM WIKI 패턴 (2026년 봄 공개)

LLM Wiki 3층 — Raw → Wiki → Queries, 그리고 file back

1층 · 불변

Raw Sources

원본 자료 (PDF·URL·전사본)
절대 수정하지 않는다 — 이민 가방
위키 페이지의 출처로 남는다

2층 · AI가 유지

Wiki

AI가 원본을 읽어 합성한 마크다운
10~15개 교차참조 페이지로 증분 갱신
모순은 플래그 · index.md가 묶는다

3층 · 합성 답변

Queries

AI와의 대화 결과
좋은 답은 Wiki로 file back ↔
질의가 수집처럼 복리로 누적

위키는 사람이 쓰지 않습니다. **AI가 쓰고 유지합니다.** 사람은 무엇을 모으고 무엇을 물을지를 책임집니다.

- 같은 구조를 도서관 언어로 옮기면

여러분이 평생 해오신 일 그대로입니다.

1층 · 불변

원자료 서고

논문 PDF·웹클리프·메모. 절대 수정하지 않는다. **사람이** 수집

2층 · AI가 유지

정리된 주제 파일

개념·요약·상호참조. 자라고 서로를 참조한다. **AI가** 작성

3층 · 근거와 함께

참고봉사

자연어 질문에 출처 링크와 함께 답한다. **사람이 질문**

달라지는 것은 하나입니다. **분류와 연결의 노동을 AI가 맡습니다.** 무엇을 모으고 무엇을 물을지는 여전히 사람의 몫입니다.

- 분류 체계가 아니라 흐름입니다 — 자료는 한 방향으로만 이동합니다

네 개의 층 — 00 → 10 → 20 → 30

00 · 입구

Inbox

웹클리프·PDF·메모가 일단 떨어지는 곳. 분류하지 않는다 — 속도가 전부.

10 · 불변

Raw Sources

논문 원문·전사본의 불변 보존. AI가 읽는 1차 소스이자 모든 인용의 최종 근거.

20 · AI 유지

Wiki

AI가 쓰고 유지하는 주제 파일. 서로를 참조하면서 자란다.

30 · 산출

Queries

질문의 답이 페이지로 남는 곳. 논문도 강의도 글도 여기서 나온다.

십진분류처럼 '어디에 넣을지' 고민하는 구조가 아니라, **지금 어느 단계인지만 보면 되는 구조**입니다.

- AI가 1만 개 노트에서 관련된 것을 꺼내는 원리

메타데이터가 하는 일 — 사서가 가장 잘 아는 그 일

type

무엇인가

논문노트·개념·인물·질의결과. 타입이 있어야 "개념 페이지만 훑어봐" 같은 지시가 작동한다.

description

라우팅 테이블

한 줄 설명이 AI에게는 색인이다. 1만 개 중 관련된 3-4개를 꺼내는 판단이 여기서 일어난다.

source · citekey

출처 사슬

이 페이지가 어느 원자료에서 왔는지. 인용 추적과 연구윤리 공개 요건이 이 필드에 달려 있다.

related · cites

그래프의 뼈대

사람이 손으로 하던 상호참조를 기계가 읽을 수 있는 형태로 만든 관계 필드.

결국 메타데이터 설계가 시스템의 성능을 결정합니다. 그리고 이건 사서가 평생 해오신 일입니다.

- 이 규칙 전체를 AI가 읽는 문서로 명문화하면

개인지식관리 스키마 — 9개의 시스템 파일

기술 규칙

**CLAUDE · AGENTS ·
ANTIGRAVITY**

AI가 볼트를 만지기 전에 읽는 작업
규정 — 표기법·메타데이터·출력 경로

철학 · 분류

CMDS · Guide · HQ

100-900 분류 체계 + 표준·템플릿
+ 네비게이션 — 장서개발정책과 분
류표

시각 언어

DESIGN

산출물이 따르는 브랜드·시각 규약

페르소나

**BRAIN ·
BRAIN_PROMPT**

나를 대신 답하는 AI의 근거 문서와
응대 규칙

사서 언어로 — **편목 규칙 + 장서개발정책 + 참고봉사 매뉴얼**을 AI가 읽을 수 있게 쓴 것. 검색이 정확한 이유는 모델이 아니라 이 규칙에 있습니다.

- 업계는 이것 'AI-READY DATA'라고 부릅니다 — 깨끗한 데이터가 아닙니다

사람 중개 없이 AI가 바로 쓸 수 있는 상태 — 다섯 가지 조건

① 통합

한 시야로

흩어진 소스가 모이고 충돌이 해소된다.
→ 입구를 하나로, 모순은 플래그

② 정제

일관되게

타입·중복·빈값 처리가 의도적이다.
→ 노트 타입 계약 + 린트

③ 의미 표지

뜻이 붙어야

`acct\_st`가 아니라 "고객 계정 상태(일 단위 검증)".
→ frontmatter의 한 줄 설명

④ 거버넌스

끝까지 따라

권한·출처 이력·동의를 임베딩 이후에도 유지된다.
→ source·citekey + 시스템 파일 규칙

⑤ 발견 가능

찾을 수 있어야

설명·소유자·신선도가 카탈로그에 있다. 시스템이 못 찾으면 ready가 아니다.
→ index.md + 갱신일

한 조사에서 데이터가 "완전히 AI-Ready"라고 답한 조직은 7%였습니다. 병목은 모델이 아니라 데이터 기반입니다 — 그리고 이 다섯은 여러분이 백 년 해온 일의 이름입니다.

- 규칙과 색인이 있어도, 실제로 찾아주는 기관이 따로 있습니다

로컬 검색 인덱스 — 내 컴퓨터 안의 목록 시스템

서가 단위

컬렉션 14개

폴더별로 인덱스를 나눈다. "위키만"
"논문 분석만" 같은 범위 지정 검색이
가능해진다 — 분관별 목록.

색인 작성

로컬 임베딩

문서를 좌표(벡터)로 바꾸는 모델이
내 컴퓨터에서 돈다. 한국어 대응 모
델, 외부 API 호출 없음.

목록 DB

SQLite + 벡터 인덱스

파일 한 개짜리 데이터베이스에 어휘
색인과 벡터 색인이 함께. 파일이 바
뀌면 그 문서만 증분 갱신.

최종 판단

재순위(rerank)

1차로 넓게 건진 뒤 질문과의 적합도
로 다시 줄 세운다. 참고사서가 대출
대에서 하는 그 판단.

현재 제 인덱스: **10,738건 · 14개 컬렉션** (마더십 9,090 + 위키 볼트 1,648). 전 과정이 로컬 — **자료가 밖으로 나가지 않습니다.**

- "검색"은 하나가 아닙니다 — 묻는 방식이 다섯 가지

무엇을 어떻게 찾는가 — 세 가지 질의, 두 가지 점검

어휘 · BM25

키워드로

단어 빈도로 점수. 용어·고유명사가 확실할 때. **키워드 검색**

의미 · 벡터

뜻으로

표현이 달라도 의미가 가까우면 찾는다. **주제 검색**

가설 · HYDE

답을 상상해서

"이런 답이 있을 것"을 먼저 쓰고, 그와 닮은 문서를 찾는다. 어휘를 모를 때 강하다.

분포 · GREP

전수 점검

정확한 문자열이 어디에 몇 번. 누락·표기 흔들림 점검. **서지 정비**

구조 · 그래프

지형 보기

문서 묶음을 노드·군집으로. 중심 주제와 다리 역할 문서가 보인다.

셋을 함께 던지고 결과를 합칩니다. 하나로 부족한 이유는 사서라면 아십니다 — 이용자는 자기가 찾는 것의 이름을 모릅니다.

● 왜 하필 마크다운인가

지식의 수명은 내용이 아니라 그릇이 결정합니다.

감히는 그릇

죽은 파일

특정 프로그램이 없으면 안 열린다. 회사가 사라지면 형식도 사라진다. 10년 전 파일을 지금 열어보십시오

열린 그릇 · FILE-OVER-APP

살아있는 파일

메모장으로도 열린다. 10년 뒤에도 열린다. 그리고 AI가 가장 잘 읽는 형식이다

실용적으로도 유리합니다. **공짜이고, 벤더에 갇히지 않습니다.** 예산 압박 속 도서관에 이건 작은 문제가 아닙니다.

- 이 이야기를 드리는 개인적인 이유가 하나 있습니다

기록은 사람보다 오래 남아, 공공의 자산이 됩니다

이어령 · 기록한 사람

서재가 곧 집필실

초대 문화부 장관, 저서 199종. 1980년대부터 워드프로세서로 쓰고 말년까지 여러 대의 컴퓨터로 집필한 디지로그의 실천 — 서재의 책 6,400여 권과 컴퓨터 7대가 그대로 보존돼 있다.

강인숙 · 보존한 사람

기억을 버려지지 않게

영인문학관 관장. 문인들의 원고·편지·초상화 3만여 점을 수집·보존했다. 개인의 서재를 공공의 아카이브로 바꾼 반세기 — 『글로 지은 집』의 저자.

그리고 그 집

글로 지은 집

두 분이 평창동에 지은 집에서 영인문학관이 태동했다. 올해 겨울, 제가 그 집에 들어가 삽니다 — 기록으로 살아온 부부의 집에서, 기록을 가르치는 일을 이어갑니다.

개인의 기록이 서재가 되고, 서재가 아카이브가 되고, 아카이브가 문학관·도서관이 됩니다. 오늘 지은 그릇도 그 계보 위에 있습니다.

● 분명히 하고 넘어가겠습니다

경쟁이 아니라 총위가 다릅니다.

기관의 검색 서비스

강물

Primo Research Assistant · Scopus AI 등. 이미 여러 도서관 검색대에 들어와 있는 훌륭한 기관 도구. 재직 중의 인프라

연구자 개인의 지식 베이스

각자의 우물

연구자가 소유하고 졸업·이직 후에도 들고 간다. 평생의 인프라

둘은 만나야 합니다. 오늘은 그중 아직 비어 있는 쪽, 개인의 우물을 짓는 이야기입니다.

- 논문 한 편이 지식이 되기까지 — 실제 화면으로

실물 파이프라인 — 다섯 단계

1 · 수집

Zotero

논문 PDF와 서지. 서지는 레퍼런스 매니저에, 해석은 볼트에.

2 · 문헌노트

@citekey

볼트의 노트가 citekey로 Zotero와 연결된다.

3 · 원자화

Paper Ingest

논문 1편 → 12개 분석 좌표 → 30-100+ 지식 원자. 인용·수치는 원문 그대로.

4 · 검증

기계 대조

모든 인용을 원자료와 대조. 전부 통과해야 완료.

5 · 질의

파일백

근거와 함께 답하고, 좋은 답은 다시 위키로 쌓인다.

요약이 아닙니다. 내 논문 쓸 때 원래의 수치와 인용문을 원문 그대로 꺼내 쓰기 위한 구조입니다.

- 쓰시던 ZOTERO를 버리지 않습니다 — 그 위에 분석 층을 엮습니다

논문 해부 — citekey에서 지식 원자까지

연결

citekey 참조

Better BibTeX가 발급한 키로 참조만. 서지·DOI·PDF는 Zotero가 계속 관리 — 중복 저장 없음.

해부

12개 분석 좌표

인용정보→연구질문→이론틀→표본→측정→분석→결과→한계→후속→서술가치. 논문 유형 6종별 변주.

산출

30-100+ 지식 원자

원자 하나 = 주장 하나 + 원문 인용 + 페이지. β값·조작적 정의·표 수치가 verbatim 보존.

검증

기계 대조

모든 인용문을 원자료와 문자 단위 대조. 환각 인용을 사람의 성실성이 아니라 구조로 막는다.

해당 없는 좌표는 이유를 적고 건너뛹니다. 억지로 채우지 않습니다.

● 논문만이 아닙니다 — 기사·리포트·정책 자료도 같은 파이프라인

수집에서 지식망까지 — 다섯 단계

1 · 클리핑

마크다운 저장

웹 클리퍼로 긁으면 원문 URL·수집일이 함께 박힌다.

2 · 고정

Raw 불변 보존

원문 페이지가 사라져도 사본이 남는다 — 링크 부패 대비.

3 · 컴파일

위키 갱신

AI가 개념·인물·기관 페이지를 쓴다. 1건 → 10-15개 페이지가 움직인다.

4 · 교차참조

모순 플래그

기존 자료와 어긋나면 표시 — 사람이 못 알아채는 충돌을 기계가 잡는다.

5 · 인덱스

로그 갱신

무엇이 들어오고 바뀌었는지 기록. 지식망 전체가 자란다.

파일 하나가 늘어나는 게 아니라 **지식망 전체가 조금씩 자라는** 구조입니다.

● 쌓기만 하면 창고가 됩니다 — 도는 구조여야 지식이 됩니다

세 동작이 순환합니다 — 흡수 · 질의 · 정화

1 · INGEST — 흡수

넣으면 퍼진다

원본을 불변으로 고정하고, AI가 읽어 관련 페이지들을 **증분 갱신**한다. 새 페이지를 하나 만드는 게 아니라 기존 10~15개가 함께 자란다.

2 · QUERY — 질의

묻고, 남긴다

근거와 함께 답하고, 좋은 답은 파일로 남아 다음 질문의 재료가 된다. 소비가 아니라 **축적**이 되는 지점.

3 · LINT — 정화

스스로 점검한다

모순·낡은 주장·고아 페이지·빠진 상호참조·언급만 되고 없는 개념을 찾아낸다. **장서 점검을 기계가 매번** 도는 셈.

이 세 동작이 돌기 때문에 시간이 갈수록 정확해집니다. 방치하면 낡는 폴더와 갈리는 지점입니다.

- AI가 1만 개를 다 읽지 않습니다 — 목록부터 읽습니다

카탈로그와 수불부 — 두 개의 파일이 전부를 조율한다

INDEX.MD

콘텐츠 카탈로그

페이지 목록 + 한 줄 요약. 질문이 오면 AI가 **이 파일**을 가장 먼저 읽고 어디를 펼칠지 정한다. 열람목록 그 자체.

LOG.MD

시간순 수불부

무엇이 언제 들어오고 무엇이 바뀌었는지 **덧붙이기만** 한다. 지우지 않는다 — 자료의 이력이 곧 신뢰의 근거.

점진적 공개

목록 → 본문

목록에서 후보를 좁히고, 필요한 문서만 펼쳐 읽는다. 전부 밀어넣지 않으니 **정확하고, 싸고, 빠르다.**

사서라면 익숙한 순서입니다. 서가로 바로 가지 않고 목록을 먼저 봅니다. AI에게도 같은 순서를 지키게 하는 것이 설계의 핵심입니다.

- 붙여넣기와 그릇의 차이

RAG는 소비, 위키는 자본

휘발성 · 붙여넣기

매번 재발견하고 폐기

질의마다 검색·합성·응답·폐기. 결과가 어디에도 남지 않는다. 다음 질문은 다시 백지에서 출발한다

누적 · 컴파일 위키

한 번 합성하고 계속 쌓임

합성 결과가 파일로 남는다. 교차참조는 이미 거기 있고, 좋은 답은 다시 위키가 된다. 물을수록 자산이 커진다

정직하게 말씀드리면, 이 복리는 **자료가 쌓였을 때** 발현합니다. 초기엔 세션 도구가 편할 수 있습니다. 결론은 하이브리드 — 다만 **남는 쪽에 그릇을 두십시오.**

- 오전 기초의 '에이전트 너머' — 그 너머에 있는 것 하나가 기억입니다

에이전트의 기억도, 파일로 수렴하고 있습니다.

에이전트 메모리 제품군

'내가 누구인가'

선호·이력·개인화를 기억하는 제품들. 2026년 업계는 마크다운 파일 기반으로 수렴 중

연구 지식 베이스

'무엇이 참인가'

근거·출처·사실을 기억하는 그릇. 열어보고, 고치고, 지울 수 있는 파일 — 기록관리 언어로, 평가·폐기가 가능한 기억

에이전트는 기억 위에서만 작동합니다. 연구자의 에이전트에게 그 기억이 곧 오늘 지은 지식 베이스입니다. 옴시디언 사용자는 이미 도착해 있습니다.

- 지식 베이스가 서면, 논문 작성은 그 위에서 일어납니다

AI가 대신 쓰는 게 아니라, 내 자료 위에서 조립됩니다.

문헌 리뷰

종합

쌓인 선행연구를 주제별로. 무엇이 **빠져 있는**지가 보인다

인용 · 출처

추적

각 주장이 어느 원자료에서 왔는지 링크로 남는다

초고

조립

개요 → 섹션 초안 → 근거 연결. 판단은 연구자가

사서는 이 여정의 **설계자이자 안내자**입니다.

- 출판계의 공통 원칙 — AI는 저자가 될 수 없다 + 사용을 공개하라

공개하려면, 과정이 기록돼 있어야 합니다.

ELSEVIER

별도 선언 섹션

원고 말미 Declaration · AI 이미지는
공개 조건부 허용 (2026-08 개정)

SPRINGER NATURE

Methods 기재

LLM 사용 문서화 · 리뷰어의 원고 업
로드 금지

연세대

공개 원칙 제시

연구윤리 가이드라인 — AI 활용 공개

한밭대

체크리스트 의무

학위논문 제출 시 자가점검

채팅창에 흡어진 사용은 공개할 방법이 없습니다. 지식 베이스가 곧 연구윤리 인프라입니다.

- 먼저 하신 도서관들이 있습니다 — 오늘 제안은 그 다음 단계입니다

다음 학기 이용자 교육 모듈 — '그릇 짓는 법'

1주차 · 프레임

그릇의 원리

연구자료가 흩어지는 이유와 그릇의 원리 — 마크다운
· 문헌노트·위키의 3층.

2-3주차 · 파이프라인

직접 지어보기

Zotero → 문헌노트 → 위키 컴파일 실습. 수강생 각
자의 연구주제 하나로.

4주차 · 논문 연결

새 참고봉사

질의·근거 추적·AI 사용 공개 요건. 사서가 'AI 사용 공
개 안내'라는 새 참고봉사 영역을 갖는다.

'붙여넣는 법'이 아니라 '그릇 짓는 법'을 가르치는 교육입니다.

● 90일 정착 로드맵

개인 생산성 도구가 곧 이용자 서비스의 원형입니다.

①

혼자

작은 주제 하나로 시작. 내 업무자료·관심 논문 10편이면 충분하다

②

팀 공유

부서 안에서 구조를 공유. 무엇이 걸리는지 실무로 확인

③

서비스화

대학원생·연구자 대상 이용자 교육 프로그램으로. 도서관의 새 참고봉사

여러분이 먼저 써보지 않은 것을 이용자에게 가르칠 수는 없습니다. 순서가 중요합니다.

- "우리 도서관은 지금 어디쯤인가" — 그걸 재는 자를 만들었습니다

AKM Index — 지식 시스템을 재는 자

무엇을 재나

5필러 × 25기준

노트가 아니라, AI 에이전트가 지식 기반을 읽고 쓰고 유지하는 시스템의 성숙도. 100점 만점

어떻게 재나

증거 없으면 점수 없음

기준마다 실물 아티팩트 요구 · 급조 산출물 불인정 · 반복 가동은 30일 증거

결과

M0-M4 성숙도 밴드

점수와 함께 가장 약한 필러가 나온다. 순위표가 아니라 다음 한 걸음의 지도

정직 공개

제 시스템도 M3

3년 운영한 볼트도 만점이 아니다 — 가장 낮은 축은 반복 사이클. 자가 진단 공개

평가는 **전 과정 로컬**에서 돌아갑니다. 보드 제출은 선택이고, 안 하시면 아무 기록도 남지 않습니다.

- 전부 오픈소스입니다

AI에게 자료를 붙여넣지 말고, 그릇을 지으십시오.

다운로드해서 오피디언으로 열면 바로 시작됩니다.

내일 여러분 도서관의 대학원생 교육에 그대로 쓰셔도 됩니다.

도서관의 역할을 다시 묻는다면 — 학생과 연구자가 자기 지식 인프라를 짓게 돕는 곳입니다.

수집·분류·연결·검색은 원래 여러분의 전문성이었고, AI 시대에 그것은 프리미엄이 됩니다.

 labs.cmdspace.work/keris-0827 · github.com/johnfkoo951/cmds-llm-wiki · cmdspace.work



labs.cmdspace.work/keris-0827

오늘 자료 · 스타터킷 · 후속 링크

감사합니다 · 질문 받겠습니다

- 질문 받겠습니다

무엇이든 편하게 물어보십시오.

볼트를 여러 개로 나누는 문제 · 검색 방식 선택 · 기관 도입 절차 · 개인정보와 데이터 주권 — 심화 주제도 좋습니다.

우리 조직의 지식 관리 수준을 재보고 싶으시면, 공개 운영 중인 자가진단 루브릭이 있습니다.



akm.cmdspace.work