

판단전용 AI 모델 Jev 입문

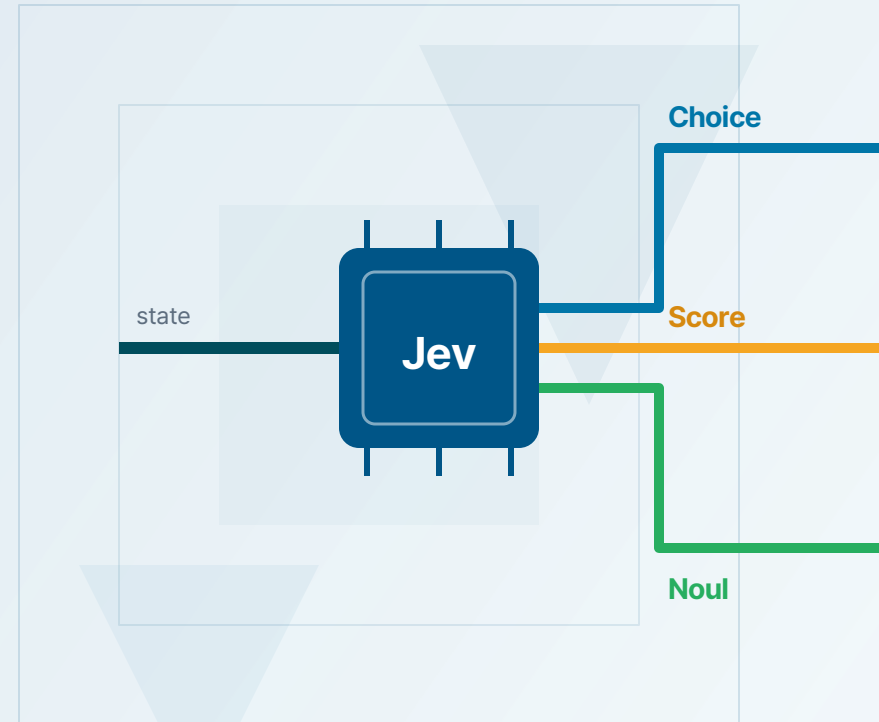
"ChatGPT가 말 잘하는 비서라면, Jev는 의사결정권자"

PROJECT CODE

20261001-판단형 A...

DATE

2026.10.01



Agenda

01

등장 배경과 핵심 개념

채팅용 LLM의 판단 병목에서 출발한 Jev의 정의와 이름 유래

02

작동 원리와 판단 유형

입출력 구조와 Choice·Score·Noul 세 판단 유형의 작동 원리

03

TypeSafe AI와 모델 신뢰성

TypeSafe AI의 창업 배경과 투자, 벤치마크로 본 신뢰성

04

활용 사례와 적용 패턴

라우팅·대량 처리·실시간 판단·에이전트 가드, 네 분야의 적용 사례

05

한계와 도입 전략

LLM과 Jev의 역할 분담 기준과 약점 영역

06

오픈소스 대안과 생태계

JevBench 순위와 하드웨어 요구량으로 본 대체 모델 후보



KEY FOCUS

판단형 AI 모델 Jev 입문

SECTION 01

등장 배경과 핵심 개념

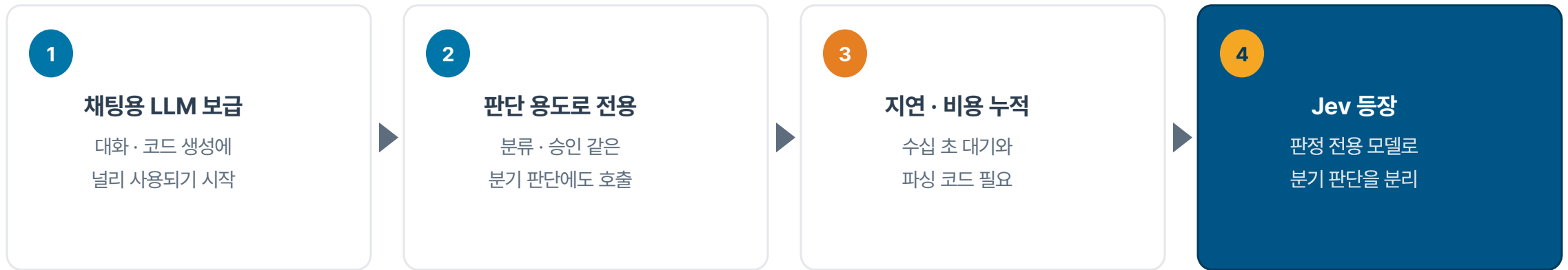
채팅용 LLM의 판단 병목에서 출발한 Jev의 정의와 이름 유래



KEY INSIGHT

채팅용 LLM의 분기 판단 오용이 지연·비용 문제를 낳아, 판정 전용 모델 Jev가 등장

채팅 · 코드 생성용 LLM이 널리 쓰이면서, 같은 모델에 분기 판단까지 맡기는 사용 방식이 퍼졌습니다.
문의 분류 · 명령 승인처럼 예 / 아니오면 충분한 판단에도 생성형 LLM 호출이 쓰이기 시작했습니다.



TypeSafe AI, 2026년 9월 — 판정 전용 모델 Jev 공개

ChatGPT 공동 발명자가 세운 TypeSafe AI가 생성형 LLM과 달리, 분기 판단만을 위해 설계한 모델을 선보였습니다 — 대화나 코드 생성이 아니라 선택과 점수를 반환합니다.

Source: velog.io · therearenorules.co.kr (2026-09 공개 기준)



모든 일을 한 에이전트에 맡길 때 함께 커지는 병목

01 · 등장 배경과 핵심 개념



KEY INSIGHT 네 병목의 공통 원인은 생성과 판정을 한 모델이 모두 떠안는 구조 —
모델을 키워도 비용 · 지연 · 승인 대기가 같이 늘어나 해결되지 않음



담당 부서 하나를 고르려고 수십 초를 기다리는 LLM 호출

01 · 등장 배경과 핵심 개념

익숙한 대기 장면



Claude Code

테스트 수정을 맡기고
추론 과정을 수십 초 지켜보는 대기



ChatGPT

긴 글을 붙여 넣고 답이
조금씩 흘러나오기를 기다리는 대기



직접 배포한 에이전트

요청 1건에 수십 초가 걸려
타임아웃

사실은 판단만 필요한 호출



문의 도착

"청구가 이중으로 됐어요"



판단
어떤 건인가?



청구 건 → 경리

결함 건 → 개발

그 외 → 영업

같은 모양의 판단들

Claude Code 가 실행하려는 명령이 위험한가 — 예 / 아니요

도착한 문의의 담당은 누구인가 — 청구 · 기술 · 영업

그러나 LLM 에 물으면

매번 정중한 설명 문단으로 답이 돌아오고
원하는 값을 뽑는 파싱 코드가 따로 필요



KEY INSIGHT 분기 판단에 생성형 LLM 을 쓰면 지연 · 비용 · 파싱 부담이 한꺼번에 붙음 —
판단만 필요한 호출이 자동화 전체 속도의 병목



토큰마다 모델 전체를 다시 돌리는 LLM 순차 생성

01 · 등장 배경과 핵심 개념



전부 다 쓸 때까지 대기

그 뒤에야 JSON 파싱 · 검증 시작



수 초~수 분 소요

출력이 길어질수록 지연도 비례



형식 깨짐 · 불필요한 설명 섞임

파싱 실패 처리 코드가 별도로 필요

원하는 것은 부서 이름 하나인데, 그 한 단어를 얻으려고 JSON 전체를 한 토큰씩 생성

KEY INSIGHT JSON 한 줄도 토큰 수만큼 모델을 반복 실행 —
판단용 호출에서 지연과 형식 오류가 생기는 근원



원 발표의 정의

텍스트를 생성하지 않고 판단 결과를 돌려주는 AI

선택 · 채점 · 확률을 소프트웨어 안에서 사용

공식 정의 — System One 모델

비정형 상태를 받아 타입이 정해진 확률적 결정을
단일 병렬 패스로 반환

공식 사양 (docs.typesafe.ai/models, 2026-09-29 확인)

항목	내용
제공사	TypeSafe AI (샌프란시스코) 2026-09-15 오픈 액세스 공개
현행 모델	jev-1.13.0 별칭 jev-latest · jev-preview 모두 1.13.0 을 가리킴
요금	입력 100만 토큰당 \$0.042 출력 토큰은 과금하지 않음
컨텍스트	요청당 64k 토큰 상태와 가장 긴 질문의 합은 32k 이하
입력	텍스트 전용 (문자열 · JSON · 텍스트 배열) 이미지 · 음성 · 영상은 받지 않음
출력	선택지 최대 255개 중 하나와 확률 자유 문자열은 생성하지 않음 (공식 블로그 명시)

분기까지 가는 경로 비교



파싱 단계가 없어 형식 오류를 처리하는 코드가 빠짐

공식 지연 70~500ms · 프런티어 모델 대비 40~200배 빠름

TypeSafe 자체 평가 — 독립 검증 아님

KEY INSIGHT Jev 의 출력은 코드가 바로 비교 · 분기할 수 있는 값 — 생성 문장을 파싱하고 형식 오류를 처리하던 코드를 줄일 수 있음



KEY INSIGHT 출력이 스키마 밖으로 나가지 못해 파싱과 재시도가 불필요 — 다만 타입이 맞아도 값은 틀릴 수 있어 정확도 보장과는 별개

일반 LLM의 출력

- 사람이 읽을 문자열을 출력
- 프로그램이 받으려면 파싱 · 검증 · 재시도 필요

Jev의 출력

- 미리 정한 선택지 안의 결정만 반환
- 결정마다 보정된 확률을 함께 반환

회사가 공개한 제약

- 컨텍스트 3만 2,000 토큰
- 선택지 최대 255개
- 문자열 생성 불가, 판단 이유 설명 없음

회사 이름

TypeSafe

소프트웨어 공학의 타입 안전성(type safety)에서 유래
반환값이 정해진 종류(숫자 · 날짜 · 범주 · 예/아니요) 안에 있음을 보장

모델 이름

Jev

19세기 경제학자 윌리엄 스탠리 제번스의 이름에서 유래
효율이 오르면 소비가 오히려 늘어나는 제번스 역설을 가리킴

모델 부류

System One

대니얼 카너먼이 말한 빠르고 직관적인 사고(시스템 1)에서 유래
깊은 추론 대신 빠른 판단을 맡는 모델이라는 뜻



타입이 정해진 답과 확률만 돌려주는 System One Model

01 · 등장 배경과 핵심 개념



정의 — System One Model

비정형 상태를 입력받아 문자열이 아닌
타입이 정해진 확률적 결정을 돌려주는 모델 부류

Jev 가 첫 모델

역할 — 프로그램 안의 판단 부품

사람에게 말을 거는 조수 대신
코드 흐름 속에서 판단만 담당

별칭 smart if-statements(똑똑한 if 문)
해설 글 · 커뮤니티에서 널리 쓰이나 공식 블로그 본문에서는 미확인

KEY INSIGHT 문자열을 한 글자도 만들지 않으므로 파싱 실패 · 형식 깨짐이 구조적으로 없음 —
대신 선택지 밖의 답은 처음부터 불가능해 질문 설계가 곧 품질



KEY INSIGHT

지금까지의 LLM · 추론 모델은 모두 느린 숙고(System 2) 쪽을 흉내 냄 —
반사처럼 즉시 끝나야 하는 판단은 말길 모델이 없었음

대니얼 카너먼 “생각에 관한 생각”(Thinking, Fast and Slow)의 사고 구분



System 2 — 느리고 신중한 사고

예시 수학 문제 풀이, 코드 작성, 긴 보고서 검토

특징 단계를 밟아 추론, 시간과 비용이 큼

AI Claude · GPT 같은 LLM과 추론 모델

글을 한 토큰씩 만들며 생각을 전개



System 1 — 빠르고 직관적인 판단

예시 “저 떡볶이는 빨간색” 처럼 즉시 나오는 판단

특징 생각을 늘어놓지 않고 곧바로 결론

AI Jev — **TypeSafe** 의 첫 **System One** 모델

글 없이 보기 · 점수 · 예/아니오와 확률만 반환

영상 발표자의 비유

지금까지의 System 2 AI 는 뇌에 해당하고, Jev 는 처음 나온 반사신경에 해당

새 도구보다 새 역할의 AI



KEY INSIGHT

Jev 는 사람의 선호보다 판단 확률의 정확도에 맞춰 학습 —
채팅 모델의 과신 문제를 모델 학습 단계에서부터 겨냥

학습 방식에 따라 갈라진 모델 계보



RLCD: Reinforcement Learning for Calibrated Decisions

TypeSafe 와 창업자

창업자 디오고 알메이다(Diogo Almeida)

Google Brain · OpenAI 연구자 출신

OpenAI 에서 RLHF 연구에 참여 — 텍스트 생성기를

대화가 통하는 채팅 모델로 바꾼 학습 기법

출시와 투자

공개일 2026-09-15

투자 4천만 달러 유치 동시 발표

정의 소프트웨어 안의 결정을 위한 첫 공개
System One 모델, 자동화에 최적화



KEY INSIGHT

분류형 판단은 Jev가 빠르고 저렴하나 정답률은 별개 — 배수 광고가 아닌 실측 정확도로 판단

분류 한 번에 수 초를 쓰는 LLM 과 1초 미만인 Jev 의 판단 경로

문제 제기 — 분류 · 예/아니오 판단에도 LLM 이 문장을 생성해 JSON 으로 파싱 → 느리고 비싸며 파싱 실패 · 과잉 확신 발생

항목	프런티어 LLM · System 2	TypeSafe Jev · System 1
역할	자유 기술 · 깊은 추론 · 코드 생성	미리 정한 선택지 위의 구조화 판단 (분류 · 점수 · 예/아니오)
생성 방식	토큰을 하나씩 잇는 자기회귀(Autoregressive) 생성	단일 병렬 패스 문자열 생성 없음
지연	3초 ~ 329초 (측정 조건 · 모델 미기재)	종단 70~500ms 프런티어 대비 40~200배 빠름 (공식) 정정 — p.2 300초 · p.3 329초 표기 불일치
단가	입력 100만 토큰당 \$0.20 ~ \$10 (모델 미기재)	입력 100만 토큰당 \$0.042 출력 무료 (공식 444.6배는 벤더 특정 사례) 정정 — 자체 수치로는 4.8~238배
구조화 오류	0.58% ~ 45.5% (출처 미기재)	스키마 위반 0 — 선택지 밖 응답 불가 (정답률과는 별개)
한계	형식 강제 없이 무엇이든 생성	문자열 생성 불가 · 선택지 255개 · 컨텍스트 64k



TypeSafe 자체 평가

항목	기존 LLM	Jev
학습 방식	사람 선호 최적화 RLHF Reinforcement Learning from Human Feedback	보정된 결정 최적화 RLCD Reinforcement Learning for Calibrated Decisions
돌려주는 것	문자열 — 파싱 필요	타입이 정해진 값과 확률
생성 방식	토큰 하나씩 순서대로	한 번 호출로 동시에
속도	3~329초	0.07~0.5초
요금 (100만 토큰)	입력 0.2~10달러 출력은 입력의 약 5배	입력 0.042달러 출력 0달러
자기 확신	물어봐도 과대	Choice · Score 답에 확신도 동봉 원 발표의 모든 답 → Noul 은 확신도 없음 (정정)

RLCD 는 논문 · 보상 함수 · 데이터셋이 공개되지 않은 벤더 고유 용어 (MindStudio, 2026-09-16) · 주황 음영 = 자사 비교 수치

💡 **KEY INSIGHT** 속도 · 요금 행은 TypeSafe 자사 비교표 수치 —
우리 업무 문장으로 다시 재기 전까지는 도입 근거가 아니라 가설



System One

대니얼 카너먼(Daniel Kahneman) · Thinking, Fast and Slow

시스템 1

빠르고 직관적

위험을 느껴 피함

표정을 읽음

시스템 2

느리고 논리적

계산

차분한 비교

책에서 시스템 1은 성급한 결론을 내는 쪽

TypeSafe 의 주장

확률 보정으로 그보다 믿을 만한 빠른 판단 가능

Jev

윌리엄 스탠리 제번스(William Stanley Jevons) · The Coal Question(1865)

증기기관 효율이 오르자 석탄 소비가 줄지 않고 증가



싸지면 수지가 맞지 않던 용도가 새로 생김



TypeSafe 의 전망

지능도 싸지면 같은 경로로 쓰임새 확대

KEY INSIGHT 이름에 담긴 가설은 판단이 싸지면 쓰임새가 늘어난다는 전망 —
호출 단가가 도입 범위를 정하는 변수라는 뜻



KEY INSIGHT

LLM에 맡기는 범위가 좁을수록, 확률성이 결과 품질에 미치는 영향도 함께 좁아진다 — 재현성 확보.

항목	전 — LLM이 본문·수치 생성 확률성이 품질을 좌우	후 — 결정적 렌더러 + 좁은 LLM 역할 확률성을 구조로 가둠
수치·사실 출처	LLM 생성 (환각 위험)	결정적 스크립트가 실데이터로 채움
같은 입력 재현성	낮음 — 실행마다 변동	높음 — 표·수치 동일
LLM 역할 범위	넓음 (사실 + 서술)	좁음 (서술·문장 다듬기)
모델 성능 의존도	높음 — 프론티어급 필요	낮음 — 로컬 소형 모델로 충분
감사·대조	어려움	쉬움
대표 위험	가짜 수치 사고	서술 표현의 경미한 편차

파급 사실 생성 책임이 스크립트로 넘어가면 로컬 소형 모델(gpt-oss·Gemma 4)로도 실무 품질 → 10장 경제성으로 연결

SECTION 02

작동 원리와 판단 유형

입출력 구조와 Choice·Score·Noul 세 판단 유형의 작동 원리



신뢰도에 따라 자동 처리와 사람 검토로 나뉘는 판단 흐름

02 · 작동 원리와 판단 유형

KEY INSIGHT

확실한 판단만 자동 처리하고 애매한 건만 사람에게 전달 —
사람이 모든 건을 보던 루프가 예외 건만 보는 루프로 축소

같은 고객 문의 — 이 문의를 어느 팀이 처리해야 하나?

LLM 에게 물으면

“문의 내용을 보면 지금 실패가 핵심이라
결제 쪽 문제로 보이며, 다만 연동 오류일
가능성도 있어서 기술팀 확인도...”
긴 글 · 확신 정도를 알 수 없음

Jev 에게 물으면

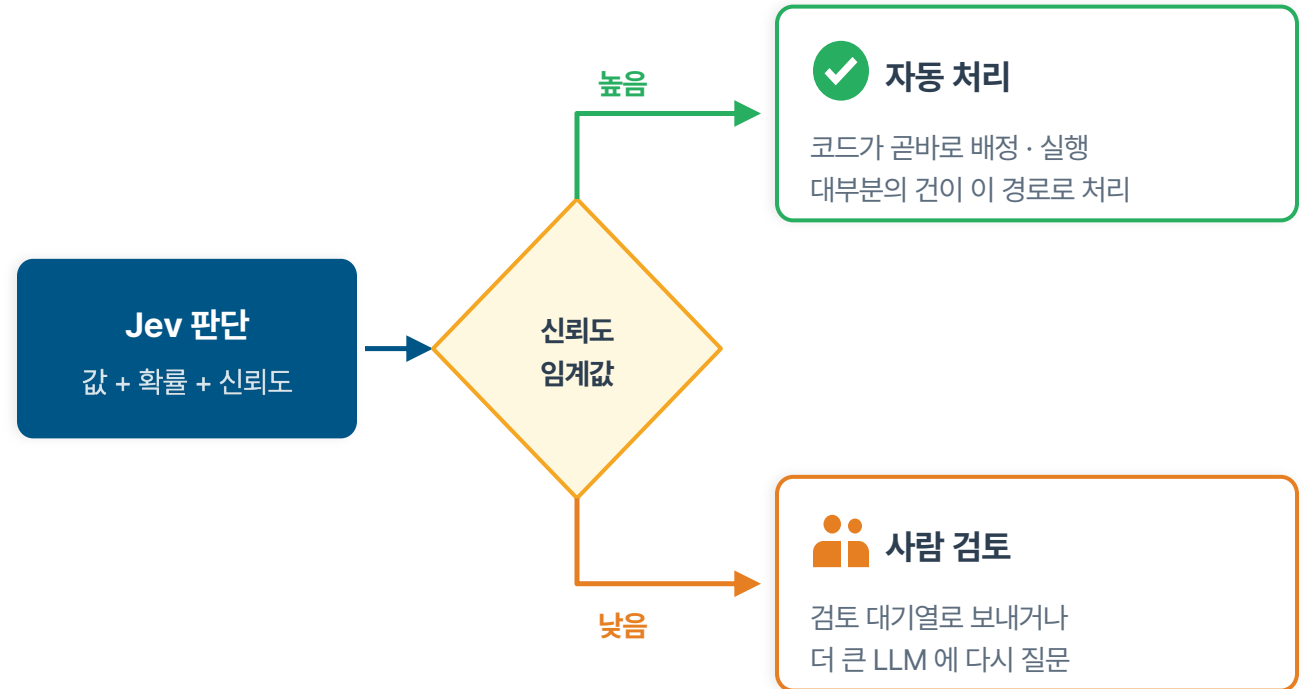
결제팀 100%

영상 시연의 답 — 보기 이름과 확률 한 줄

보기 밖의 답이나 설명 글 없음

확률을 보고 다음 동작을 코드가 결정

신뢰도 임계값으로 나뉘는 처리 경로



임계값은 호출하는 쪽 코드에서 설정 — 업무 위험도가 클수록 높게 조정



KEY INSIGHT

Jev 출력은 파싱 없이 분기 조건으로 직결 — 해석 오류·지연은 사라지나 선택지는 사람이 미리 정의해야 함

문자열 없이 형식이 정해진 확률만 돌려주는 System One 모델 Jev

모델 정보

개발사

TypeSafe AI (샌프란시스코)

2026-09-15 얼리 액세스 · 시드 4천만 달러(DCVC)

이름

System One — 카너먼의 빠른 사고(시스템1)

Jev — 경제학자 제번스(W. S. Jevons)

학습

RLCD — 보정된 확률이 목표

논문·데이터 미공개, 벤더 고유 용어

사양

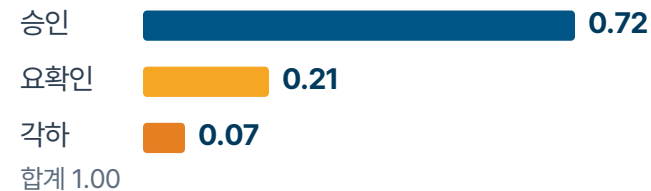
jev-1.13.0(별칭 jev-latest) · 요청당 64k 토큰

텍스트 입력 전용 · 한국어 지원(영어보다 부정확)

POST /v1/systemone · client.system_one(state=..., questions={...})



예시 분포 — Choice 질문(선택지 3개) 결과



판본 선택 — 무관한 점수 6개 대신 합이 1인 일본어판 분포 채택, 공식 Choice 형태와 일치

정정 — 일본어판 0~100% 표기를 공식 문서 척도 0~1 로 통일



질문 유형마다 모양이 달라지는 Jev 의 확률 출력

02 · 작동 원리와 판단 유형

	Choice (선택)	Score (점수)	Noul (Yes / No)
돌려주는 값	정의한 선택지 전체의 확률 분포 선택지 최대 255개	말로 기술한 루브릭 위의 점수와 분포	명제가 참일 확률 하나 (0~1)
확신도	있음	있음	없음
용도	담당자 · 모델 라우팅	긴급도 · 리스크 정량화	실행 허가 · 승인 게이트 트리거
예시 일본어판	A안 60% B안 25% C안 10% D안 5%	저 중 고 극고 0 · 0.25 · 0.5 · 0.75 · 1 구간 경계는 사용자가 루브릭으로 정의	Yes 0.73 0.27 No 는 1 - 0.73 Noul 은 Bernoulli 에서 딴 TypeSafe 조어

확신도 — 분포가 한 답에 몰리면 높고 퍼지면 낮은 0~1 통계 (공식 문서 예시 코드 기준)

0.5 미만 — 사람에게 넘김	0.5 ~ 0.9 — 확인 · 추가 정보	0.9 초과 자동
------------------	------------------------	-----------

정정 원 자료 p.5~6 의 모든 답에 확신도가 붙는다는 설명 → Noul 은 제외 (공식 문서)

판본 선택: Score 예시는 영어판 3단계(low / medium / high) 대신 일본어판 4구간 채택

KEY INSIGHT 확신도는 Choice · Score 답에만 붙고 Noul 답에는 없음 —
Noul 로 게이트를 만들면 확신도가 아니라 Yes 확률 자체에 문턱을 걸어야 해 설계 방식이 달라짐



switch 문과 임계값 비교, if 분기에 하나씩 대응하는 세 판단 유형

02 · 작동 원리와 판단 유형



Choice

어느 것인가

예시 용도

요청에 맞는 AI 모델 고르기

반환 필드

choice · probabilities · confidence

코드 대응

후보별 코드 경로로 분기 (switch)

제약

선택지 최대 255개



Score

어느 단계인가

예시 용도

풀 리퀘스트(PR, Pull Request)의
리뷰 필요도 채점

반환 필드

score · legend · probabilities · confidence

코드 대응

점수를 임계값과 비교해 분기

제약

단계는 2~10개, legend 로 단계 설명 반환



Noul

그렇다고 말할 수 있는가

예시 용도

사내 RAG(검색 증강 생성) 답변의
인용 근거 확인

반환 필드

noul (Yes 확률 0~1 하나)

코드 대응

확률 문턱을 넘으면 if 분기

제약

confidence 없음 — 확률 자체가 판단 값
이름은 Bernoulli 에서 온 조어

공식 사용 지침 — 한 질문에 한 가지 즉답 판단만. 요인이 여럿이면 요인별로 따로 묻고 코드에서 가중치로 결합



KEY INSIGHT 답의 모양에 맞는 유형을 고르면 결과를 파싱 없이 분기에 바로 사용 —
정도를 묻는 판단을 Noul 로 물으면 0.5 가 중간 수준으로 오독되어 분기가 틀어짐



질문마다 반환 필드가 갈리는 Jev 의 세 판단 형

02 · 작동 원리와 판단 유형

형	묻는 것 (예)	돌아오는 값
Choice	어느 담당 창구인가	선택지 1개 선택지별 확률 분포 confidence
Score	우선도가 어느 단계인가	소수 점수 (예 1.66) 단계별 확률 분포 confidence
Noul	데이터 유실 우려가 적혀 있는가	참일 확률 0~1 하나 confidence 없음 이름은 Bernoulli 에서 온 조어

응답 예 — 공식 퀵스타트 축약

공식

```
{
  "answers": {
    "department": {
      "type": "choice",
      "choice": "technical",
      "confidence": 0.78,
      "probabilities": {
        "technical": 0.85,
        "billing": 0.15,
        "sales": 0.0
      }
    },
    "frustration": {
      "type": "score",
      "score": 1.0,
      "confidence": 1.0,
      "probabilities": {
        "0": 0.0,
        "1": 1.0,
        "2": 0.0
      }
    },
    "is_urgent": {
      "type": "noul",
      "noul": 1.0
    }
  }
}
```

is_urgent 에는 confidence 필드가 없음 — Noul 은 확률 하나만 반환

Choice 선택지 상한 255개 (공식 블로그 · Choice 문서)

KEY INSIGHT Noul 에만 confidence 가 없음 —
보류 문턱을 하나로 통일할 수 없어 질문 유형마다 사람에게 넘길 기준을 따로 정해야 함



KEY INSIGHT

확신도는 Choice·Score 전용 값 — Noul은 확률값 자체에 문턱 걸어야 분기 성립

문자열 대신 분기 조건에 바로 쓰는 값을 돌려주는 Jev 출력 타입

Choice 다중 선택

선택지 중 하나를 선택

반환: choice · 확률 · confidence

집중형 — 확신도 높음

0.94

billing 0.94 · technical 0.03

sales 0.02 · other 0.01

분산형 — 확신도 낮음

billing 0.34 · technical 0.30

sales 0.22 · other 0.14

* 확률 합은 항상 1.00 — 원자료 147% 예시는 정정

Score 단계 평가

순서 있는 루브릭 위의 점수

반환: score(보간값) · 단계확률

· confidence · legend

0 기준 미충족

1 일부 충족

2 완전 충족

1.035

확신도 92% — 원자료 예시

점수 = 단계 × 확률의 가중합

→ 단계 사이 값도 가능

Noul 확률적 진위값

명제가 참일 확률 0.0~1.0 값

반환: noul 값 하나 — confidence 없음

명제

입력 문장에 역사적 날짜 오류 있음

P(Yes)

0.99

원자료 예시 — Jev에 불리한 과업

→ 분기는 확률값에 직접 문턱 설계

(다음 장 3분기 제어 참고)



입력 state

이 문장의 오타를 고쳐 줘



질문 questions.route

```
{
  "type": "choice",
  "instructions": "적절한 모델은?",
  "criteria": {
    "fast": "오타 수정 · 단순한 추출",
    "powerful": "복잡한 설계 · 원인 분석"
  }
}
```

필드 읽는 법

instructions — 무엇을 판단할지 적는 질문

criteria — 후보 이름과 각 후보의 판단 기준

choice — criteria 키 가운데 하나만 반환

후보가 입력을 다 덮지 못하면 other 후보를 따로 둬

반환 answers.route

가상 · API 미실측

```
{
  "choice": "fast",
  "probabilities": {
    "fast": 0.95,
    "powerful": 0.05
  }
}
```

fast → 경량 모델 호출

실제 응답에는 type · confidence 도
함께 옴 (위 반환은 발췌)



독립 평가의 주의 사항

고객 문의 분류 실험에서 확신도가 낮은 것을

더 큰 모델로 넘기자 정확도가 **0.7%p 하락**

확신도는 사람에게 넘길 시점의 신호로 쓰는 편이 근거 있음

SENZO-NCEKANA/jev-triage-benchmark · 합성 200건



KEY INSIGHT 고른 값이 criteria 키 안에서만 나와 라우터 코드가 예외 처리 없이 분기 —
후보가 입력을 다 덮지 못하면 other 후보를 따로 뒤야 억지 분류를 막음



단계별 확률의 가중 평균으로 나오는 PR 영향 범위 점수

02 · 작동 원리와 판단 유형

입력 state

공통 API 변경을 포함한 PR 의 설명 · 변경 내용(diff) · 테스트 정보



질문 questions.impact

```
{
  "type": "score",
  "instructions": "변경의 영향 범위는?",
  "criteria": [
    "표시 · 문구만 변경",           ← 단계 0
    "1개 기능의 동작을 변경",      ← 단계 1
    "여러 기능 · 공통 처리를 변경" ← 단계 2
  ]
}
```

공식 문서가 밝힌 점수의 성질

두 단계 사이 값이 나올 수 있음 (예: 1.43)

서로 다른 분포가 같은 점수를 만들 수 있음

공식 예: $0 \times 0.0 + 1 \times 0.57 + 2 \times 0.43 = 1.43$

점수와 confidence 를 함께 보고 리뷰어 배정

반환 answers.impact

가상 · API 미실측

```
{
  "score": 1.6,
  "probabilities": {
    "0": 0.1, "1": 0.2, "2": 0.7
  }
}
```

실제 응답에는 legend(단계 번호 → 설명) · confidence 도 포함 — 위는 발췌

막대 폭 = 단계별 확률 · 아래 축 = 0~2 점수 위치



$$0 \times 0.1 + 1 \times 0.2 + 2 \times 0.7 = 1.6$$

criteria 배열 순서(0부터)에 확률을 곱해 더한 가중 평균

💡 **KEY INSIGHT** 같은 1.6 점도 확률 분포에 따라 뜻이 달라짐 —

점수 하나로 리뷰어를 배정하면 두 단계 사이에서 걸린 PR 을 놓치므로 confidence 를 함께 확인



인용이 붙어 있어도 원문과 어긋난 RAG 답변을 거르는 Noul 확률

02 · 작동 원리와 판단 유형

인용 원문 — 사내 절차서

개발 환경은 전원, 운영 환경은 운용 담당자만

두 텍스트를 함께 state 로 전달하고, 주장이 인용 원문으로 뒷받침되는지 한 번 묻는 구조

질문 questions.supported

```
{
  "type": "noul",
  "instructions": "주장은 인용 원문으로 뒷받침되는가?",
  "criteria": {
    "true": "인용 본문이 주장을 뒷받침함",
    "false": "인용 본문과 모순 · 근거 부족"
  }
}
```

noul 값 읽는 법

0 에 가까움 — 강한 No

0.5 근처 — 불확실

1 에 가까움 — 강한 Yes

▲ 이번 답변 0.08

💡 KEY INSIGHT 출처 표기만으로는 답변 정확성을 보증하지 못함 —

표시 직전 Noul 한 번으로 모순 답변을 재생성 경로로 돌리면 틀린 절차가 사용자에게 노출되기 전에 차단

⚠ RAG 답변 — 인용과 모순

전원이 운영 환경에 배포할 수 있습니다 [사내 절차서]

반환 answers.supported

```
{
  "noul": 0.08
}
```

설명용 · API 미실측

Yes 확률

0.08

→ 표시 전에 재생성

criteria 는 선택 항목 — true · false 키로 Yes · No 의 뜻을 고정

공식 Noul 형식과 일치 · 별도 confidence 는 반환하지 않음



같은 0.5 라도 뜻이 갈리는 Score 값과 Noul 값

02 · 작동 원리와 판단 유형

Score 정도를 잴

단계를 먼저 정의

0 보통

1 이르게

2 최우선

결과는 0~2 사이 소수 — 단계별 확률의 기대값

1.66 은 이르게와 최우선 사이



0.5 는 보통과 이르게 사이 — 정도의 크기



Noul 질문은 참 · 거짓을 가릴 수 있는 평서 명제로 작성

"우려가 있나?" 보다 "우려가 명시되어 있음" 처럼 쓰는 편이 해석이 명확



KEY INSIGHT Noul 0.5 는 중간 정도가 아니라 예 · 아니오 반반 — 이 값을 등급처럼 읽으면 임계값 설계가 통째로 어긋남

Noul 명제의 참거짓을 잴

질문은 명제 하나

"데이터 유실 우려가 문의에 명시되어 있음"

결과는 그 명제가 참일 확률

0.5 는 판단이 반반인 상태



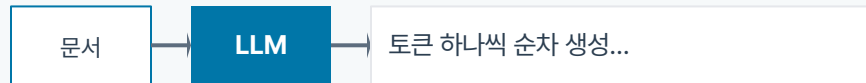
0.5 는 확신 없음 — 중간 정도라는 뜻 없음



KEY INSIGHT

질문을 한 호출로 묶는 병렬 구조가 속도의 원천 — RLCD 보정 학습은 미공개라 자체 검증 필요

토큰 순차 생성 — LLM

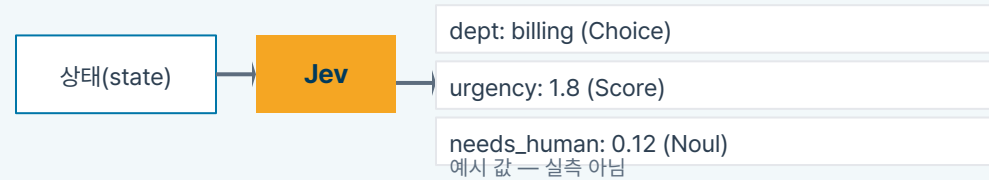


매 토큰마다 모델 전체를 다시 실행

지연이 토큰 수만큼 누적

응답이 끝난 뒤에야 파싱 가능

단일 병렬 패스 — Jev



질문 여러 개를 한 요청에 담아 한 번에 수신

출력은 타입이 정해진 확률적 결정값

typesafe-sdk · POST api.typesafe.ai/v1/systemone · jev-latest

RLHF

Reinforcement Learning
from Human Feedback

사람이 선호하는 답에 맞춰 최적화

RLVR 류

검증 가능한 보상 기반
강화학습

정답을 기계로 확인 가능한
보상에 최적화

RLCD

Reinforcement Learning for
Calibrated Decisions

예측 확률을 실제 적중 빈도에
맞춰 보정 최적화

미확인: 논문 · 보상함수 · 데이터셋
· 보정곡선 비공개

정정 — 원 자료 classifier.invoke → system_one · 수십ms → 종단 70~500ms



상황과 질문을 한 번 읽고 모든 답을 동시에 내는 Jev 호출

02 · 작동 원리와 판단 유형



세 답이 동시에 반환 · 발표자 실측 예시



답은 반드시 선택지 안에서 반환

출력 토큰 과금 없음 · 질문을 늘려도 호출 횟수는 그대로



KEY INSIGHT 질문 수가 늘어도 호출은 한 번 —

질문을 요청 여러 개로 쪼개는 설계보다 한 요청에 묶는 설계가 지연 · 비용 모두 유리



방식	내용	Jev와의 차이
 Structured Output 구조화 출력	JSON 스키마에 맞춰 생성	여전히 생성 방식이라 속도 거의 동일
 전용 분류 모델	직접 학습	학습 데이터 · 재학습 운영 필요
 Jev	말로 묻고 확률로 받음	학습 없음 · 병렬 · 확신도 포함



보강 — 커뮤니티의 반론

Hacker News에서는 BART 제로샷 분류 같은 기존 접근과 비슷하다는 지적도 존재 (29장)



KEY INSIGHT

학습 데이터를 모으지 않고도 당일 시험에 들어갈 수 있는 판단 수단 —
분류기 구축 전 검증 단계의 비용과 기간 단축



입력 종류가 달라도 같은 형식으로 돌아오는 판단 결과

02 · 작동 원리와 판단 유형

발표자 실측

상황	물은 것	돌아온 답
 Slack 메시지 "내일 15시 정례 회의, 30분 미룰 수 있을까요?"	의도는? 답장이 필요한가?	일정 조정 (1.00) 필요 (0.91)
 영수증 스타벅스 1,290엔 · 메모 "고객과 미팅"	계정 과목은? 사적 이용 의심은?	회의비 (1.00) 의심 약함 (0.23)
 코드 diff DB 삭제 보호를 false 로 바꾸는 변경	위험도는? 사람 리뷰가 필요한가?	높음 (1.8 / 2) 필요 (0.83)

세 경우 모두 같은 호출 코드 — 바뀐 것은 state 와 questions 뿐

 **KEY INSIGHT** 입력 종류마다 별도 모델 · 파서가 필요 없음 — 질문만 바뀌 여러 업무에 같은 호출 코드를 재사용



KEY INSIGHT 한 질문에 한 판단, 경계는 설명에 명시 — 계산은 코드가 말아야 결과가 설명 가능해짐

한 질문에 한 판단만 담는 Jev 질문 설계 원칙

01

한 질문에 한 판단

애매한 종합 평가를
작은 질문으로 분해

02

같은 입력은 묶음

여러 질문을 한 요청으로
보내고 코드로 합성

03

선택지의 구체적 설명

경계 조건과 다른 선택지로
보낼 경우를 설명에 명시
덧셈·나눗셈 비교는 코드 처리

나쁜 예

판단 세 개를 한 점수에 섞음

```
"overall": Score(  
  criteria=["낮음", "보통", "높음"],  
  instructions="중요성·긴급성·기술성")
```

결과를 설명하기 어려움

좋은 예

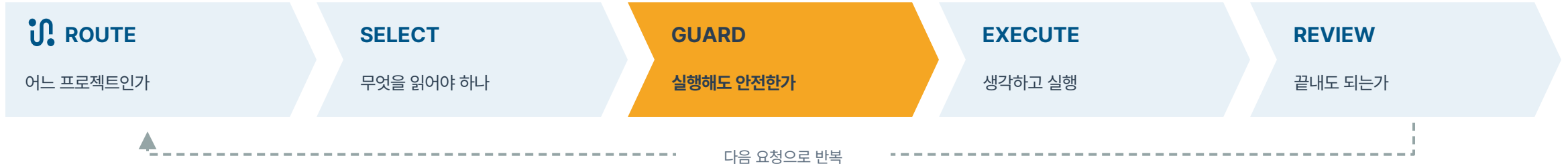
판단별로 나누고 경계를 명시

```
"team": Choice(criteria={  
  "product_info": "기능·사용법 안내",  
  "tech": "오류·장애. 사용법은 product_info"})  
# 경과일수 비교는 코드에서 처리
```

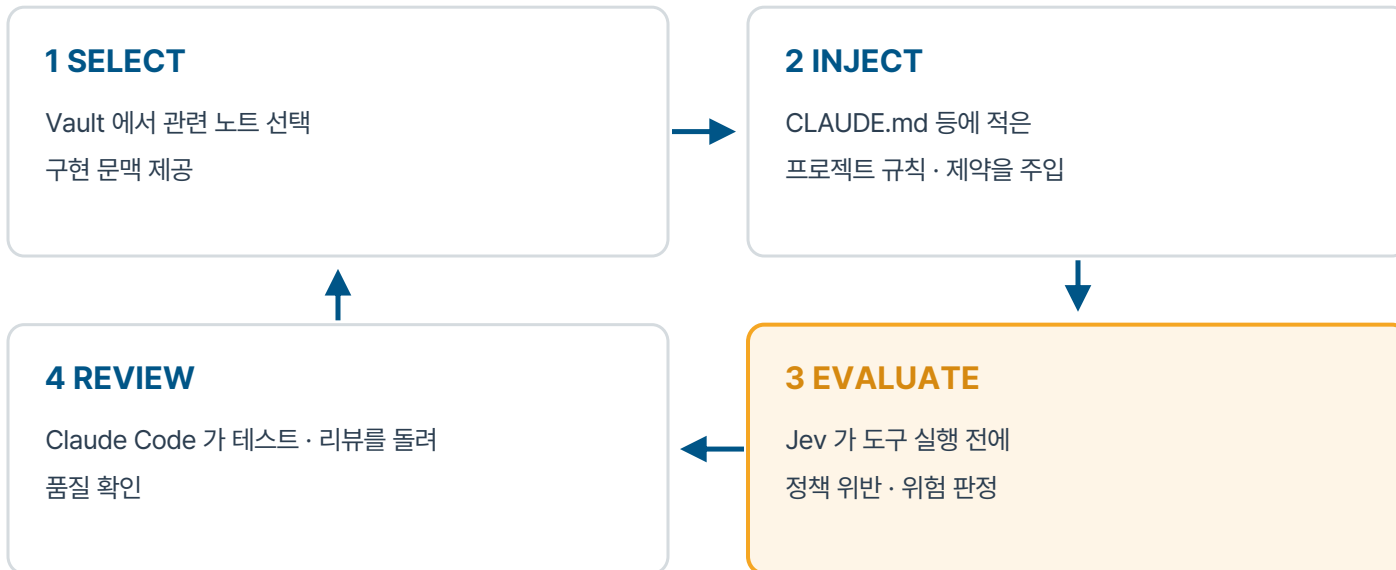
경계 사례를 설명에 직접 명시(Choice 문서)



운영 루프 — 요청 한 건마다 도는 다섯 단계



구현 루프 — Claude Code x Jev x Vault



보강 — Claude Code 공식 혹은

PreToolUse

도구 실행 전에 스크립트가
허용 · 거부를 결정하는 지점

배치 순서 (한국어판 제안)

- 1 결정적 차단 목록 (코드 규칙)
- 2 Jev 판정 (2차 확률 판정)

EVALUATE 단계가 이 혹은 안에서 실행

KEY INSIGHT Claude Code 에서 GUARD 자리는 PreToolUse 혹은 —
 도구 실행 전에 코드로 거부할 수 있는 공식 지점이 있어 Jev 판정을 붙일 위치가 이미 정해져 있음



행동마다 끼워 넣어도 부담이 작은 Jev 판정 지연과 비용

02 · 작동 원리와 판단 유형

벤더 공개 수치 — TypeSafe 자체 평가



종단 응답

70~500ms

원 자료의 수백 밀리초를 구체화



입력 100만 토큰당

\$0.042

출력 무료



자사 홈페이지 워크플로 기준

193.6배 빠름
444.6배 저렴

블로그 스스로 높은 쪽 수치라고 밝힘



요청 한도

초당 25만 토큰
분당 1,200 요청

독립 평가

평가	정확도	비용	중량 지연
고객 문의 분류 200건 합성 데이터 · 개인 저장소 · 대 GPT-4.1-mini	92.9% 대 89.9%	1,000건당 \$0.026 대 \$0.308	415ms 대 1,496ms
의료 벤치마크 arXiv 예비 결과 · 2026-09-27 · 대 GPT-6 Sol	PubMedQA 78.4% 대 78.2% — 대등 DiagnosisArena-MCQ 59.8% 대 82.4% — 크게 열세		0.31초 이하

정정 원 자료 p.7 의 압도적 저비용은 비교 기준 없음 → 공개 가격과 독립 평가 수치로 대체

 **KEY INSIGHT** 좁은 분류 · 라우팅에서는 LLM 분류기와 대등하면서 더 싸고 빠르지만 복합 추론에서는 열세 — 적용 범위를 판정 한 번으로 끝나는 질문으로 한정해야 비용 이점이 유지됨



출력 토큰을 과금하지 않는 Jev 요금 구조

02 · 작동 원리와 판단 유형

I 입력 100만 토큰당

0.042달러

10억 토큰이어도 42달러

I 출력 토큰

0달러

공식 표현 too cheap to meter
계량할 필요도 없이 저렴

💡 **KEY INSIGHT** 출력이 무료라 비용이 입력 길이에만 비례 —
상태 텍스트를 짧게 유지하는 설계가 비용 관리의 전부

홈페이지 워크플로 1,000건 처리 비용

TypeSafe 자체 평가

Jev

39센트

GPT-5.6 Luna

3.31달러

Claude Haiku 4.5

19.49달러

SiliconANGLE 보도 인용 · 막대는 비용 비례

정정 Vercel AI Gateway 표시값

원 발표 캡처 — 입력 0.04달러 (표시 자릿수 차이)

2026-09-29 Vercel 모델 페이지 — 0.042달러 / 100만 입력 토큰

공식 요금과 동일

SECTION 03

TypeSafe AI와 모델 신뢰성

TypeSafe AI의 창업 배경과 투자, 벤치마크로 본 신뢰성



공식

2026-09-15

스텔스 해제 · Jev 얼리 액세스

공식

4,000만 달러

시드 투자 · DCVC 주도

공식

jev-1.13.0

현행 모델 · jev-latest · jev-preview

TypeSafe 자체 평가

70~500ms

공식 종단 응답 시간



System One Model

비정형 상태를 읽고, 문자열 대신 타입이 정해진
확률적 결정을 단일 병렬 패스로 돌려주는 모델 부류

Jev 는 그 첫 모델

이름 유래

System One — 카너먼의 빠른 사고(시스템 1)

Jev — 경제학자 윌리엄 스탠리 제번스(제번스 역설)

판단 단가가 내려가면 판단을 부르는 곳이 늘어난다는 문제의식



TypeSafe AI

본사 샌프란시스코 · 공동창업자 3명

Diogo Almeida — CEO, 전 OpenAI,
InstructGPT 논문 공저자

Erik Gafni · Sasha Sheng

보강 — 채택 초기 지표

출시 24시간 만에 Vercel AI Gateway 유료 팀의

약 13% 가 사용 (Vercel 발표)

원 발표도 개발자 경력과 성능 평가를 분리해서 보라고 강조



KEY INSIGHT 창업자의 InstructGPT 경력은 신뢰 근거가 아니라 배경 정보 —
성능은 공개 사양과 자기 데이터 검증으로만 판단



KEY INSIGHT CEO가 자신이 만든 RLHF를 한계로 지목 — 사람의 선호에 맞춘 훈련이 과장된 답을 낳는다는 문제의식이 제품 방향의 출발점

Diogo Almeida

공동창업자 · CEO

- Google Brain을 거쳐 OpenAI 입사
- 포스트 트레이닝 팀에서 약 4년 근무
- RLHF · InstructGPT · ChatGPT · GPT-4 작업 참여
- InstructGPT 논문 공저자
- 2024년 퇴사 후 창업

Erik Gafni

공동창업자 · CTO

- 연쇄 창업자
- Ravel Biotechnologies 공동창업 (2018년, 이후 피인수)
- Invitae · Freenome 초기 직원
- 논문 · 특허 다수

Sasha Sheng

공동창업자 · COO

- Meta/FAIR 연구 엔지니어 출신
- News Feed · AI Experiences · AI Research 조직 근무
- NeurIPS · ECCV 논문 발표
- 해커톤 다수 주최

CEO 발언

RLHF는 사람의 선호에 맞추는 훈련이라 과장된 약속이 설계상 기능이 된다고 지적

미확인

Almeida의 학력, OpenAI 재직 연도, 퇴사 사유는 공개 자료에서 확인되지 않음



💡 **KEY INSIGHT** 직원 20~30명으로 추정되는 소규모 연구소 — 공개 7일째 신규 가입을 중단해 공급 안정성이 도입 검토의 변수

- 2024년 ● OpenAI를 떠난 Almeida가 Gafni · Sheng과 창업
- 2024~2026 ● 약 2년 스텔스, 합성 데이터 기반 훈련 방식(RLCD) 개발
- 2026-09-15 ● 스텔스 해제, Jev 공개, 시드 4,000만 달러 발표, Vercel 탑재
- 09-18 ● OpenRouter 등록, TechCrunch 보도
- 09-19 ● Cloudflare Workers AI 탑재
- 09-22 ● 수요 급증으로 신규 가입 중단, 한때 API 서빙 용량 소진
- 09-24 ● 10억 달러 이상 조달 협상 보도 (The Information)
- 09-25 ● Vercel 무료 기간 종료, 정가 과금 시작

회사 규모

추정

20~30명

직원 수는 회사 비공개

외부 집계: PitchBook 30명, Built In 18명

본사	샌프란시스코, 주 5일 사무실 근무
팀 출신	OpenAI · Google Brain · Meta/FAIR · Stripe · Airbnb · Plaid · Docker
비공개	매출 · 유료 고객 수 · API 호출량
안정 버전	jev-1.13.0
채용	진행 중



💡 **KEY INSIGHT** 100억 달러 가치는 협상 보도 단계 — 라운드가 닫히지 않았고 참여자도 공개되지 않아 성사 전까지 사실로 인용 불가

확정

4,000만 달러

시드 라운드 (2026-09-15 발표)

- 리드 투자자 DCVC
- 담당 파트너 James Hardiman
- 공식 발표에 이름이 나온 투자자는 DCVC 한 곳

보도

100억 달러 이상

협상 중인 기업가치 (미확정)

- 10억 달러 이상 조달 협상
- The Information 09-24 보도
- 라운드 미종결, 참여 투자자 미공개
- 시드 기업가치 약 2억 달러 (Forbes, 익명 소식통 인용)

미확인

투자자 3곳

집계 사이트에만 나오는 이름

- Horizon Venture Capital
- Spark Capital
- AWS Startups
- 보도자료와 주요 언론에서는 확인되지 않음

성사 시 변화

시드 기업가치 약 2억 달러에서 100억 달러 이상으로, 9일 만에 50배 (두 값 모두 보도 기준)

비교 사례

OpenAI 출신 Jerry Tworek의 Core Automation, 35억 달러 가치로 6억 3,000만 달러 조달 (디지털투데이)



공개 2주 만에 별 2만 개를 넘긴 Jev 커뮤니티 프로젝트

03 · TypeSafe AI와 모델 신뢰성



browser-use/jev-ultrafast

화면을 번호 붙은 요소 표로 바꿔
동작과 대상을 한 번에 예측
중앙 작업 시간 9.45초 → 7.09초
브라우저 프로토콜 호출 1,092 → 101

별 21,237



tamaratran/fast-jev-compactation

Claude Code 플러그인
요약 대신 도구 호출 · 결과를 한 번에
점수 매겨 오래된 것만 버림

별 7,162

보강



pi-jev

코딩 에이전트 톨 실행 게이트
파괴 · 비밀 유출 · 요청 범위 밖 · 피해 규모
네 질문을 한 요청 약 300ms 에 판정

발표자 소개



jevmeter

영상 자막 1문장마다 5문항을 물어
얼버무림 정도 측정
토론 1편 5,955회 판정에 5센트

발표자 소개



Emoji Jev

입력이 멈출 때마다 이모지 254개 재정렬
1회 약 0.1초 · 0.0002달러

발표자 소개



jarrodwatts/jev-trader

블록마다 약 300ms 걸리는
매수 · 매도 판단 데모

별 2,668

보강

별 수는 2026-09-29 GitHub API 조회값 · pi-jev · jevmeter · Emoji Jev 는 저장소 미확인, 원 발표 수치 그대로



KEY INSIGHT 개발자 관심은 확인됐으나 대부분 데모 단계 —
운영 적용 근거가 아니라 설계 아이디어 참고용



KEY INSIGHT

세 사례 모두 짧은 판단 반복 구간에 Jev 적용 — 별 수 증가는 관심도 지표일 뿐 운영 품질 근거 아님

9월 21일 이후 8일 사이 별 수가 1.3~1.7배로 늘어난 공개 활용 사례

GitHub 별 수 변화 · 9/21 원 발표 → 9/29 GitHub API 조회

jev-ultrafast

1.63배

브라우저 조작

9/21 약 1.3만

9/29 21,237

fast-jev-compact

1.30배

AI 작업 이력 압축

9/21 약 5,500

9/29 7,162

jev-trader

1.67배

AI 매매 판단

9/21 약 1,600

9/29 2,668

jev-voice-browser

참고

참고 · 음성 브라우저

364

(원 발표 수치 없음)

browser-use / jev-ultrafast

화면을 번호 붙은 요소 표로 바꾸고
동작과 대상을 한 번의 왕복으로 예측

중앙 작업시간 9.45초 → 7.09초

프로토콜 호출 1,092 → 101 (저장소 자체 수치)

tamaratran / fast-jev-compact

Claude Code 플러그인 — 도구 호출과
결과마다 Jev 점수를 매겨 낚은 것은
버리거나 자르고 남길 것은 원문 유지

jarrodwatts / jev-trader

Monad 블록마다 매매 판단 1회
약 300ms 마다 매수 · 매도 판단

공개 데모는 mock · dry-run

실거래 성과 근거 아님 (원 발표 각주)

정정: 별 수는 2026-09-29 GitHub API 조회 값, 9/21 원 발표 값 병기 · 배수는 원 발표 반올림 기준 근사치 · 별 수는 관심도 지표일 뿐 운영 품질 근거 아님

💡 KEY INSIGHT 저비용·저지연이 잦은 판정 호출을 가능케 함 — 합성 리뷰 1회 실측이라 일반화는 보류

정답률 1%p 차이에 지연 2.4배 · 비용 약 4.8배가 갈린 리뷰 판정

Jev 공식 요금

\$0.042

/ 입력 100만 토큰 · 출력 무료

실험 조건 — 제3자 실측 (mameli/jev-vs-luna, 2026-09-18 실행)

리뷰 1건에서 5항목 판정: 주제 · 감정 · 추정 별점 · 답장 필요 · 결함 보고

100건×3회 · OpenRouter 경유 · Luna 는 추론 없음 + 엄격한 구조화 출력 · 60초 타임아웃, 재시도 없음

지표	Jev	GPT-5.6 Luna	차이
응답 시간 (유효 응답 중앙값)	0.647초	1.556초	Jev 가 약 2.40배 빠름
비용 (1,000회 환산)	약 \$0.032	약 \$0.154	Jev 가 약 4.8배 저렴
항목별 정답률 (실패는 오답 처리)	96.13%	97.13%	Luna 가 1.00%p 높음

Jev HTTP 520 에러 1회(41.03초, 5항목 오답 처리) · 유효 응답만 보면 96.45% 대 97.13% · 5항목 모두 정답 88.33% 대 90.67%

정정 — 비용 배수는 원 발표 1/4.9 → 실측 4.8배(원값 4.88배, 표 반올림 4.81배) · 데이터는 합성 영어 리뷰 50사례×문맥 2개=100건, 모델 순위 판단엔 부적합 명시

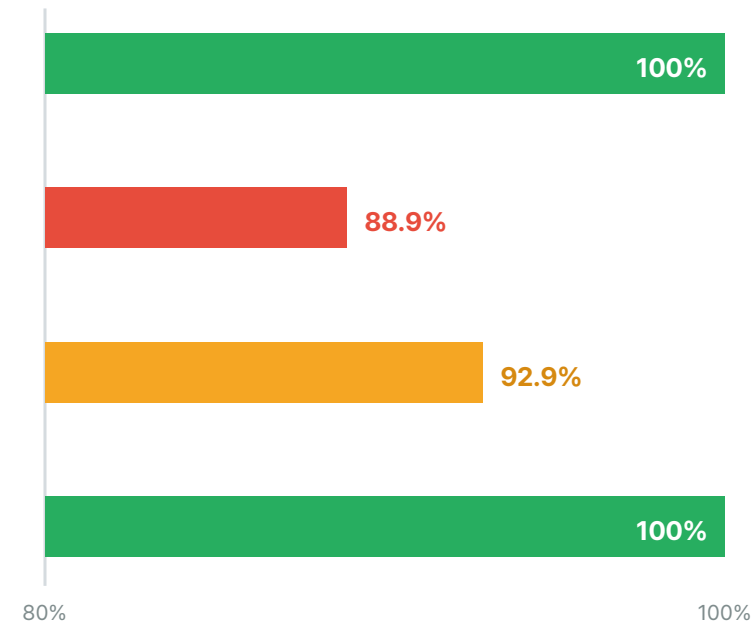


발표자 실측

작성자 라벨과의 일치 · 가상 데이터

검증 세트	라벨 일치	일치율	읽을 수 있는 점
일반 5분류	42/42건	100%	솔직한 가상 문장에서는 모두 일치
헛갈리는 5분류	16/18건	88.9%	경계가 애매한 문장에서 불일치 발생
답장 필요 여부	13/14건	92.9%	답장이 필요하다는 기준 자체에 따라 결과가 달라짐
회의록 문장 분류	12/12건	100%	원문을 그대로 둔 채 분류 가능

일치율 (80~100% 구간 확대)



전체 86건 규모의 작은 가상 데이터 — 발표자 스스로 실무 정답률로 일반화할 수 없다고 명시

공식

영어가 주력이고 한국어 등 한중일 문자는 정확도가 낮을 수 있어 운영 전 자기 데이터로 시험 권고 — 원 발표는 일본어 데이터



KEY INSIGHT

쉬운 가상 문장 100% 일치는 도입 근거가 못 됨 —

자기 업무의 경계 사례로 만든 라벨 데이터가 있어야 실제 정확도를 알 수 있음

SECTION 04

활용 사례와 적용 패턴

라우팅·대량 처리·실시간 판단·에이전트 가드, 네 분야의 적용 사례



개요 47 영상 속 활용 사례 분류 48 TypeSafe 공식 용도

4-1

라우팅·분류

요청이 들어오는 즉시 담당·모델·처리 경로를 고르는 앞단 관문

- 49 작업별 모델·담당 라우팅과 필터
- 50 장애 1차 대응 경로
- 51 고객지원 문의 라우팅
- 52 요청 난이도별 라우팅 캐스케이드

4-2

대량 평가·데이터 처리

출력 비용 없이 수백~수천만 건에 같은 기준을 적용

- 53 리뷰 수백 건 기준별 점수화
- 54 원문을 두고 라벨만 붙이는 대량 처리
- 55 리뷰 5,000만 건 Map-Reduce 태깅

4-3

실시간 판단

화면 표시·결제 승인처럼 0.1초 안에 끝내야 하는 판단

- 56 방문자별 화면 개인화와 컷 편집
- 57 결제 100ms 안의 이상거래 탐지

4-4

에이전트 보조·안전 가드

에이전트의 다음 동작을 고르고 위험한 실행·출력을 차단

- 58 에이전트 다음 동작 후보 선택
- 59 에이전트 도구 선택
- 60 도구 호출 직전 실무 가드
- 61 LLM 출력 직전 인라인 가드레일

종합 62 대량 판단을 아래층 Jev에서 끝내는 3층 Cognitive AI Stack



KEY INSIGHT

짧은 판단은 앞단·대량·실시간 모두에서 Jev 전담 — 생성형 LLM은 느려 그 자리에 안 맞음

영상 속 여섯 사례를 세 가지 쓰임으로 분류

01 앞단

요청이 들어오는 순간 판단

고객 문의 자동 분류

문의 내용을 읽고
청구·기술·영업 담당에게 바로 배정

위험 명령 사전 차단

코딩 에이전트가 실행하려는 명령의
위험 여부를 실행 전에 판정

02 대량

한 번에 수천 건을 처리

대규모 이벤트 라우팅

쏟아지는 로그·이벤트를 건별로 읽어
처리 큐에 분배

게임 NPC 행동 선택

초당 수천 번 다음 행동을
후보 중에서 골라 반응

03 실시간

지연 없이 즉답이 필요한 자리

상담 중 분기 안내

대기 없이 다음 단계를 바로 제시해
상담 흐름 유지

추천 콘텐츠 즉시 필터링

노출 직전 적합성을 판정해
지연 없이 콘텐츠 통과



라우팅

문의 · Slack · 메일을
도착 즉시 담당에게 분배



LLM 출력 채점

생성 결과를 채점해
탈옥(jailbreak) · 주제 이탈 탐지



재순위

검색으로 모은 후보를
관련도 순으로 재정렬



게임 판정 데모

TypeSafe 공식 데모 (발표자 인용)

1993년작 DOOM 을 평균 0.14초 판단으로 계속 플레이
초당 10회 판정에 1시간 약 7달러



HN 에서는 DOOM 데모가 화면 픽셀이 아닌 구조화된 게임 상태를 입력으로 써서 성능이 부풀려졌다는 지적 (29장)



KEY INSIGHT 공식 용도는 모두 짧은 판단을 대량 반복하는 자리 —
호출 빈도가 높을수록 단가 · 지연 이득이 커짐



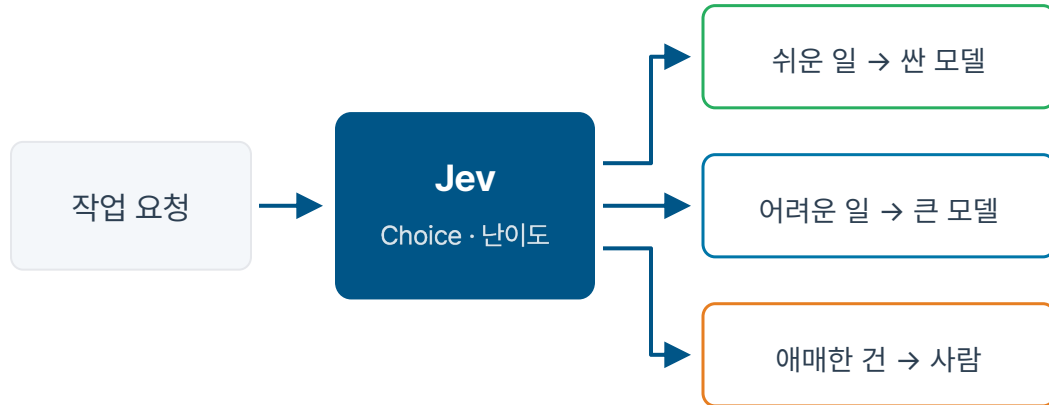
KEY INSIGHT

비싼 모델 앞에 0.1초짜리 관문을 두는 구조 —

쉬운 일은 싼 경로로, 어려운 일만 큰 모델로 보내 전체 비용 절감

사례 1 · 라우팅

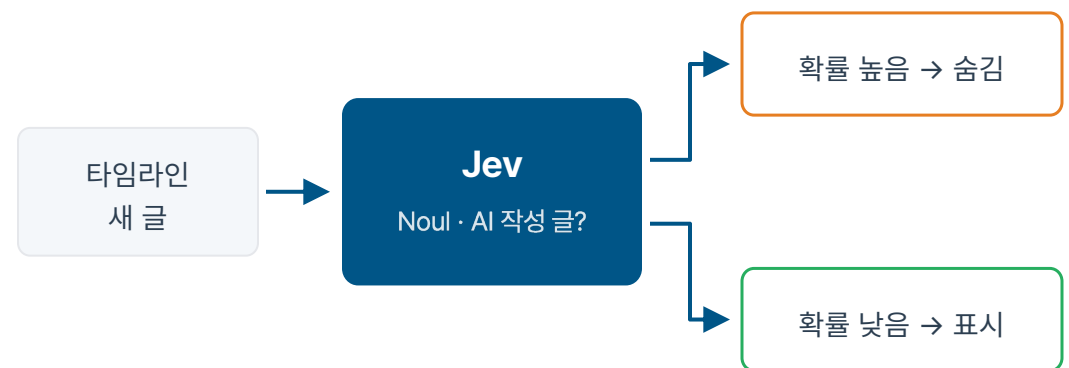
작업마다 어떤 모델에 맡길지 사람이 고르던 일을 앞단에서 판단



비싼 모델을 모든 작업에 쓰는 낭비와 싼 모델의 사고를 함께 방지

사례 2 · 필터

타임라인을 내릴 때마다 글 하나하나를 실시간으로 판별



같은 방식으로 웹페이지 광고, 댓글 스팸 · 악플 걸러내기

공식 문서의 같은 계열 패턴

인텐트 라우팅 — 요청을 규칙 · 전문 LLM · 사람으로 분배

LLM 가드레일 — 입출력 메시지를 위험 확률로 통과 · 검토 · 차단



긴급도와 담당 팀 판정으로 갈리는 장애 1차 대응 경로

04-1 · 라우팅·분류

1



Claude Code 가 MCP 로 횡단 검색

에러 로그 · GitHub Issue · Jira 티켓

MCP = 에이전트와 외부 도구를 잇는 표준 연결 규격



2



Jev 판정 라우터

긴급도(Score) + 담당 팀(Choice) 동시 평가



3a 긴급



Slack 알림

담당 팀을 곧바로 호출

3b 통상



수정 PR 초안 작성

Claude Code 가 로그로 원인을 특정하고
Draft PR 로 제출



사람 리뷰 후 병합

판본 선택 — PR 은 영어판의 초안 표현 채택,
일본어판의 자동 작성은 병합까지 자동이라는
오해를 피하려 배제



KEY INSIGHT Jev 는 경로만 정하고 수정은 Claude Code 의 PR 초안이 맡음 —
사람 리뷰를 거쳐 병합해야 모델 판단만으로 운영 코드가 바뀌는 일을 막을 수 있음

판정 질문 카드 (예시)

Score

긴급도

0.75 이상 → 긴급 경로

0.75 미만 → 통상 경로

Choice

담당 팀

확신도 0.5 이상 → 해당 팀 배정

확신도 0.5 미만 → 사람이 수동 배정

문턱값은 예시 — 공식 권장값 아님

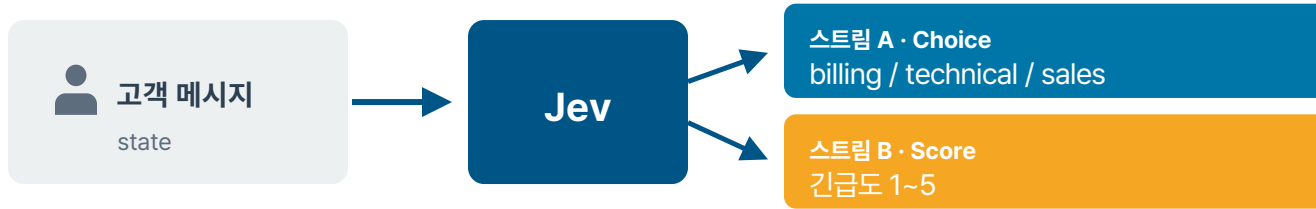
원 자료 p.15 의 Claude Code 활용 장면 중

- 로그로 버그 원인 특정 · 수정
- MCP 로 Jira 등 외부 도구 조작



부서와 긴급도를 호출 하나로 정하는 고객지원 문의 라우팅

04-1 · 라우팅·분류



분배 규칙 예시



자동 응답 봇

확신도 높음 + 긴급도 낮음



사람 상담원

확신도 중간 또는 긴급도 3~4



우선 대응 큐

긴급도 5

부서 판정 · 고객 불만 정도 · 사람 개입 필요 여부를 한 번의 병렬 패스로 취득 (원 자료)

설계 제약

처리 기한(SLA) 경과 판정은 코드가 계산 — 약점 3 날짜 비교
고객 응답 텍스트는 템플릿 또는 LLM 이 작성 — 약점 9 생성 불가
낮은 확신도는 사람에게 넘김 — 큰 모델 이관은 효과 미확인 (6번 장)



KEY INSIGHT

문의 분류는 독립 평가로 확인된 Jev 의 가장 강한 영역 —
한국어 문의는 정확도가 낮을 수 있어 실제 티켓 표본 시험이 도입의 첫 관문

원 자료 추정

비교 모델 · 측정 조건 미기재

응답 시간

8.5초 → 0.11초

티켓당 비용

\$0.01388 → \$0.000081

산술 근거: $\$0.000081 \div (\$0.042 / 100만 토큰) \approx$ 티켓당 입력 약 1,930 토큰
0.11초는 공식 중단 범위 70~500ms 안 · 월 10만 건이면 \$8.1 대 \$1,388

독립 평가

GitHub 문의 분류 벤치마크

정확도

92.9%

GPT-4.1-mini 89.9%

1,000건당 비용

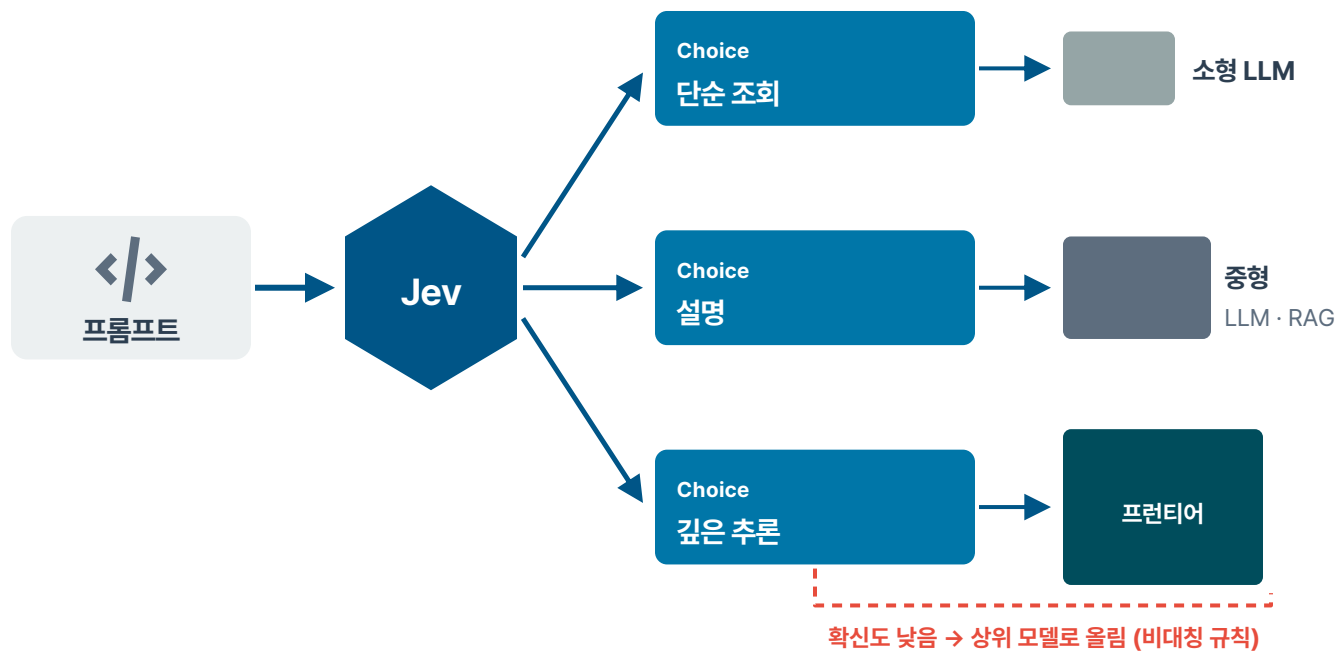
\$0.026

합성 문의 200건 · 한국어 표본은 별도 시험 필요



요청 난이도로 하위 모델 크기가 갈리는 라우팅 캐스케이드

04-1 · 라우팅·분류



설계 제약

독립 평가에서 복합 진단 20점 이상 열세 (7번 장) → 깊은 추론 여부 판정이 과소평가될 위험
낮은 확신도를 큰 모델로 넘겨도 정확도 개선 미확인 (-0.7포인트, 6번 장)
→ 라우팅 품질은 하위 모델 답의 정답률로 사후 측정

분류 작업에 프린티어 가격을 치르는 낭비를 없애는 것이 원 자료의 목적



KEY INSIGHT 난이도 판정 자체가 Jev 가 약한 복합 추론 영역 —

애매한 요청은 상위 모델로 올리는 비대칭 규칙이 없으면 어려운 질문이 소형 모델로 새는 품질 사고 위험

원 자료 추정

요청당 라우팅 비용

\$0.000025

$\$0.000025 \div (\$0.042 / 100\text{만}) \approx \text{요청당 입력 약 595 토큰}$

정정 라우팅 지연 0.64초

공식 중단 범위 70~500ms 밖의 값
네트워크 왕복이나 하위 모델 호출까지 포함한
종단값으로 추정 — 측정 조건 없음

미확인 Jev 단독 지연은 공식 범위 기준으로 표기

공식 Vercel AI Gateway 채택

약 13%

출시 24시간 내 Jev 를 쓴 유료 팀

게이트웨이 앞단 라우터 수요 확인 (Vercel 블로그, 2026-09-18)



KEY INSIGHT

건당 비용이 거의 0 이 되면서 전수 평가가 가능 —
표본만 골라 읽던 작업을 전체 데이터에 같은 기준으로 적용

★ 사례 3 · 리뷰 분석

싱가포르 호텔 수백 개 리뷰를 내 기준 세 가지로 점수화

호텔	소음	위치	청결
호텔 A · 리뷰 312건	조용함	좋음	보통
호텔 B · 리뷰 187건	시끄러움	좋음	좋음
호텔 C · 리뷰 451건	보통	없	좋음

표의 호텔 · 건수 · 등급은 형태 설명용 예시

리뷰 한 건마다 기준 세 개를 Score 질문으로 평가

모든 리뷰를 한 번씩 평가한 뒤 코드가 호텔별로 합산

영상 시연: 약 3초 만에 내 기준에 맞는 호텔 목록

같은 원리의 비즈니스 적용

- 데이터 인덱싱 — 문서마다 분류 · 태그 부여
- 마케팅 타기팅 — 고객별 관심도 점수화
- 고객 이탈 감지 — 문의 · 로그에서 이탈 징후 판별

공통점: 대량 자료에 우리 기준으로 점수 붙이기

🎯 사례 4 · 의미 기준 검색

글자가 겹치지 않아도 찾는 뜻과 맞는 내용을 선별

공식 예제 — GitHub 약관 검색

약관 218줄을 Choice 질문 하나로 관련도 점수화

Noul 질문으로 답이 문서 안에 있는지 확인

이미지 검색은 이미지 입력 지원 이후의 일 (현재 텍스트 전용)



원문을 고치지 않고 분류 결과만 덧붙이는 대량 처리 업무

04-2 · 대량 평가·데이터 처리

공식

공식 용도 · 공개 사례 개요 — 개별 사례 효과는 발표자가 재현하지 않음



업무 분배

담당 창구 분류
답장 필요 여부 판정
경비 청구의 1차 점검



대량 데이터 정리

게시물 · 논문 태그 부여
설문 자유 기술 분류



후보에서 추출

문서 안 금액 후보 중
청구 금액 고르기
회의록에 결정 · 할 일 표시

세 업무 모두 원문은 그대로 두고 라벨 · 후보만 덧붙이는 구조 — 오답이 나와도 원본 손상 없음

미검증

원 발표자의 콘텐츠 · 강의 업무 도입안 (미구현)

유튜브 댓글을 질문 · 감상 · 요청 · 오류 지적으로 분류
녹화 중 읽는 단락과 슬라이드 후보 대조

과거 게시글의 주제 · 형식 분류 후 반응 비교



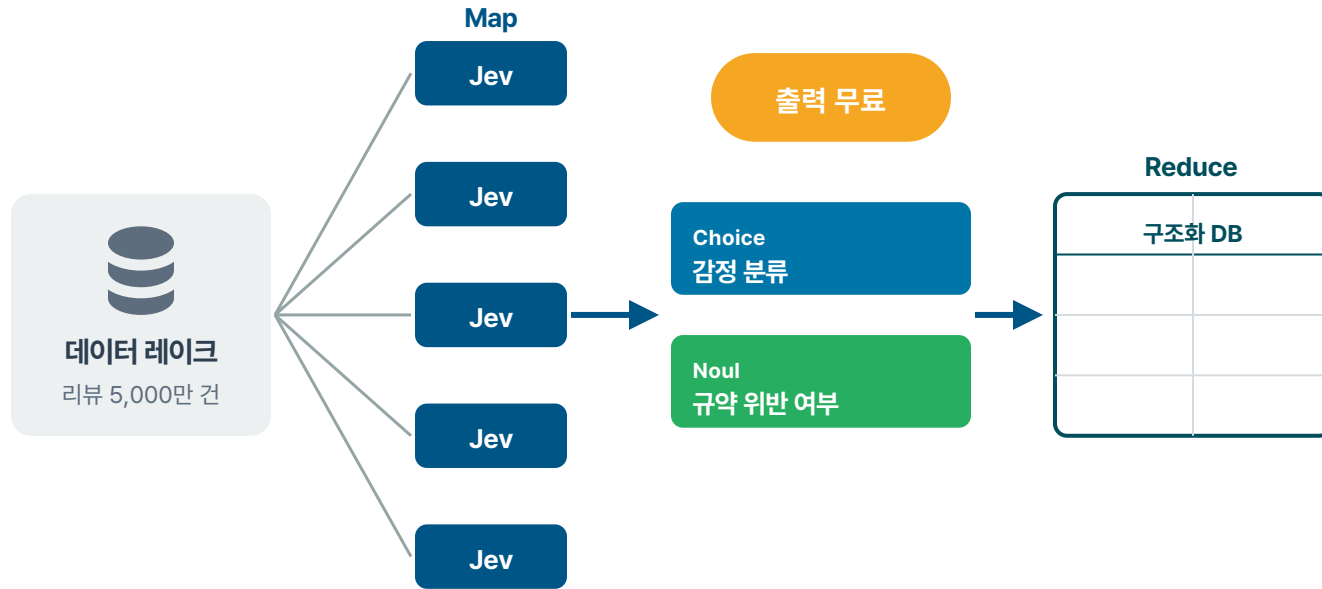
KEY INSIGHT

원문을 바꾸지 않으므로 틀려도 되돌리기 쉬운 업무부터 맞춤 —
분배 · 태그 · 후보 선별은 오답 비용이 낮아 첫 시험 대상으로 적합



출력 무료 단가로 5,000만 건을 태깅하는 리뷰 Map-Reduce

04-2 · 대량 평가·데이터 처리



대응 — 요청 수 병목 풀기

한 요청에 리뷰별 질문을 여러 개 넣는 묶음 방식 (요청당 컨텍스트 64k 토큰 안)
또는 한도 증액 협의 — 묶음 크기에 따른 정확도 변화는 미검증 (약점 5)

설계 제약

별점 · 개수 집계는 Reduce 단계의 코드가 담당 — 약점 2
한국어 리뷰는 표본 정확도부터 측정

비용 산술

원 자료 추정

가정: 리뷰당 입력 200토큰
 $5,000\text{만} \times 200 = 100\text{억 토큰}$
 $\times \$0.042 / 100\text{만 토큰}$

약 \$420

리뷰 길이에 정비례

원 자료 수백 달러 규모와 일치 · 비교 대상 LLM 비용은
모델 · 환율 미기재라 원 자료 추정으로만 표기

처리 기간 산술

공식 기본 한도 기준

요청 한도 · 분당 1,200 (리뷰 1건 = 1요청)

약 29일

$5,000\text{만} \div 1,200 \approx 41,667\text{분}$

토큰 한도 · 초당 25만

약 11시간

$100\text{억} \div 25\text{만} = 40,000\text{초}$

→ 병목은 요청 수 한도



KEY INSIGHT 대량 분석의 병목은 비용이 아니라 요청 한도 —
기본 한도로는 약 29일이 걸려 묶음 요청 방식이나 한도 증액을 먼저 정해야 일정 산정 가능

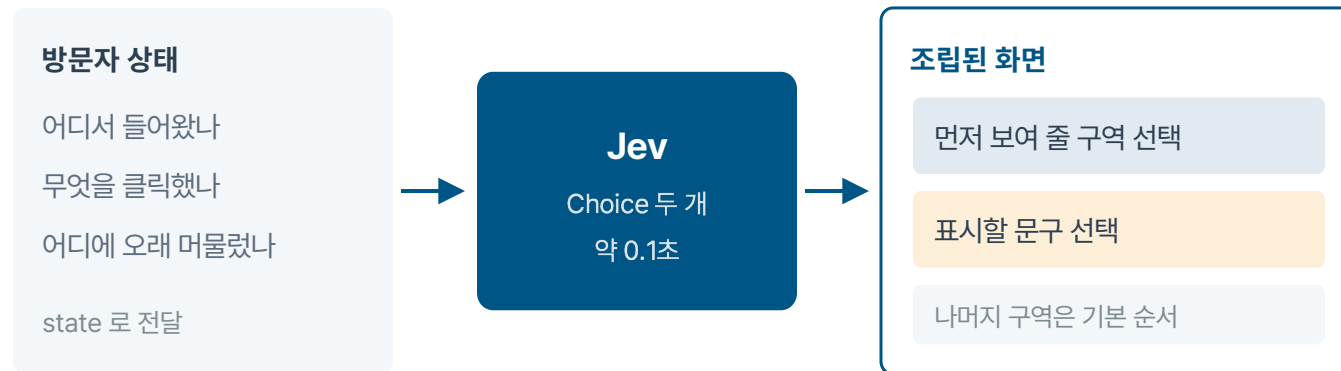


KEY INSIGHT

판단에 수 초가 걸려 불가능했던 방문자별 화면 구성이 현실적인 선택지로 전환 —
사용자는 AI 가 개입한 줄 모른 채 자신에게 맞춘 화면을 경험

사례 5 · 맞춤형 UI

모든 방문자에게 같은 화면이던 이유 — 판단을 LLM 에 맡기면 수 초 지연



방문자별 판단에 걸리는 시간

LLM  수 초

Jev  약 0.1초 — 페이지 표시 시간 안에 판단 완료

막대 길이는 시연 수치 8.566초 · 0.114초 비율 기준

사례 6 · 컷 편집

무음 구간마다 자를지와 길이를 즉시 판단



Noul · 이 무음 구간을 잘라야 하나?

Choice · 남길 길이 0.3초 · 0.5초 · 1초 중 선택

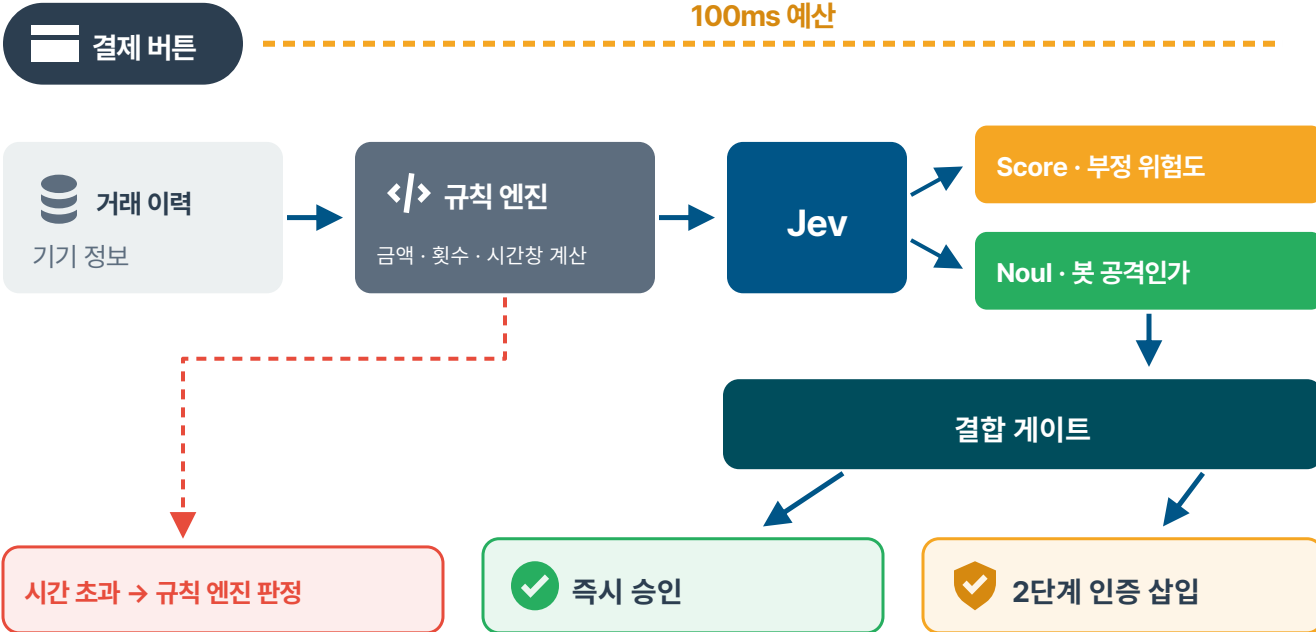
지금 도구로도 가능한 작업이지만
구간마다 판단 비용이 거의 들지 않음

질문 구성은 설명용 예시 — 음성은 텍스트 정보로 변환 필요



100ms 결제 예산 안에서 승인과 2단계 인증이 갈리는 이상거래 탐지

04-3 · 실시간 판단



판정 지연 — 두 수치 비교

원 자료 추정

공식

70~100ms

70~500ms

원 자료 값은 공식 범위의 하단만 인용한 수치
상한 500ms 는 결제 예산 100ms 의 다섯 배

→ 시간 초과 폴백이 설계의 필수 요소

처리량 산술

공식

기본 한도

분당 1,200 요청

= 초당 20건 (수요에 따라 조정 중)

결제 피크가 초당 20건을 넘는 서비스는
한도 증액 협의가 도입의 전제

설계 제약

금액 · 거래 횟수 · 최근 N분 시간차는 코드가 계산해 결과만 전달 — 약점 2 · 3

거래 메모 같은 사용자 입력 텍스트의 주입 지시 가능성 — 약점 6

→ 규칙 엔진 차단 목록이 먼저, Jev 는 2차 판정

→ 거절은 자동 실행하지 않고 2단계 인증으로 완화

원 자료의 위험도 0~100 은 Score 의 단계 보간값을 환산한 표기로 설계



KEY INSIGHT 공식 지연 상한 500ms 는 결제 예산 100ms 를 넘을 수 있음 —
시간 초과 시 규칙 엔진으로 넘어가는 폴백이 없으면 Jev 가 결제 경로의 새 장애 지점



다음 동작 후보를 고르고 실행을 코드에 넘기는 에이전트 보조 판단

04-4 · 에이전트 보조·안전 가드



조작 후보 선택

게임의 합법 수 · 화면 조작 · 다음 도구



내용 점검

자료와의 정합성 · 위험한 제안
경보 1차 판정



에이전트 보조

모델 라우팅 · 작업 상태 판정
대화 이력 선별

공개 구현 — GitHub 별 수 2026-09-29 조회

browser-use/jev-ultrafast

별 21,237

브라우저 에이전트 — 화면을 번호 붙은 요소 표로 바꿔 동작과 대상을 한 번에 예측
중앙 작업 시간 9.45초 → 7.09초 · 브라우저 호출 1,092 → 101

tamaratran/fast-jev-compaction

별 7,162

Claude Code 플러그인 — 도구 호출 이력을 점수화해 오래된 것만 버리고 나머지는 원문 유지

moritzkreimb/jev-voice-browser

별 364

음성 명령 단어마다 약 300ms 에 의도 · 대상 판정, Playwright 가 실행

alsoleg89/jev-bouncer

별 1

셸 명령 자동 허용 · 거부 가드 — 효과 검증 자료 없음



허가 범위와 실행 권한은 코드와 사람이 관리

가드로 쓸 때는 결정적 차단 목록 뒤의 2차 판정, 낮은 확신도는 사람 승인

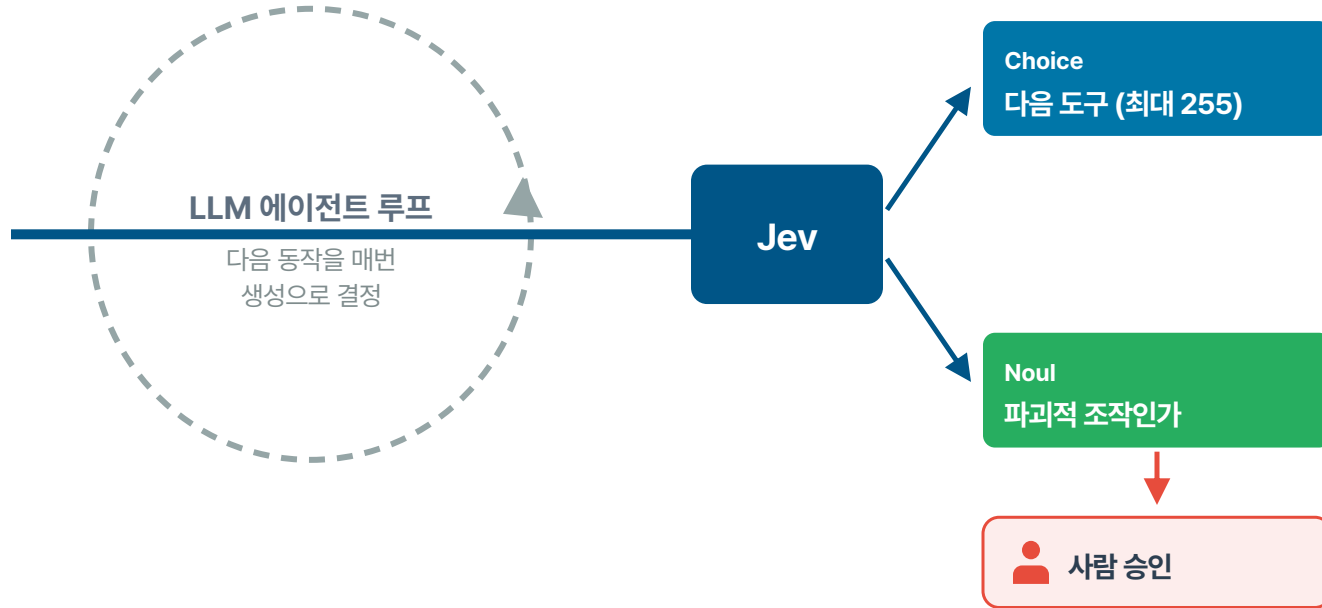


KEY INSIGHT 공개 구현은 모두 Jev 가 후보를 고르고 코드가 실행하는 구조 —
공식 약점에 주입 지시 취약이 있어 보안 판단의 단독 관문으로 쓰면 위험



LLM 루프 한 바퀴를 Jev 판정 하나로 줄이는 에이전트 도구 선택

04-4 · 에이전트 보조·안전 가드



설계 제약

도구 256개 이상이면 범주 → 도구 순서의 2단 Choice
도구 결과는 적대적 입력으로 취급 — 약점 6 → 결정적 차단 목록 뒤 2차 판정
파괴적 조작은 공식 예시대로 확신도 0.9 초과에서만 실행
관련 없는 이력은 잘라서 전달 — 약점 5



KEY INSIGHT 도구 선택은 속도 이득이 가장 크게 체감되는 자리 —
도구 결과 속 주입 지시에 약한 만큼 파괴적 조작은 확률과 무관하게 사람 승인 경로로 고정해야 안전

원 자료 추정

스텝당 판정 비용

\$0.0002

$\$0.0002 \div (\$0.042 / 100만) \approx$ 스텝당 입력 약 4,760 토큰 (도구 목록 포함)

정정 7.1초는 Jev 판단 시간이 아님

브라우저 에이전트 전체 작업 시간 — browser-use/jev-ultrafast
취리히 → 런던 항공권 검색 약 7.1초

중앙 작업시간 **9.45초 → 7.09초**

프로토콜 호출 **1,092회 → 101회**

화면을 번호 붙은 요소 표로 바꿔 입력 · 새도 루트 ·
프레임 · 캔버스 · 업로드는 범위 밖 (저장소 자체 수치)



참고 사례 jev-bouncer

Claude Code 셸 명령 허용 · 거부 판정 — 인지도 낮고 효과 검증 자료 없음



💡 KEY INSIGHT

Anthropic 분류기도 과잉행동 17%를 놓침 — 확률 가드는 차단 목록과 사람 승인을 대신 못함

도구 호출 직전에 끼어드는 Claude Code 자동 모드 분류기와 jev-bouncer

Claude Code 자동 모드 분류기

Anthropic · 2026-03-25

구조

1단계 빠른 단일 판정 → 걸린 것만 2단계 추론 재판정

오탐률

8.5% → 0.4%

1단계 단독 → 2단계 재판정 후

미탐률

실제 52건

17%

합성 1,000건

5.7%

주입 방지 설계

분류기는 사용자 메시지·실행 내용만 보고 도구 출력은 보지 않음

jev-bouncer

커뮤니티 플러그인 · 옛 이름 jev-guard

2026-09-19 생성

구조

PreToolUse 훅에서 셸 명령·파일 편집·MCP 호출 판정
셸 3분류(읽기전용·되돌릴 쓰기·파괴적) + 위험 5개

결정 순서

1. 결정적 차단 목록 우선
2. 파괴 0.10 미만 · 위험 전부 0.40 미만 → 허용
3. 파괴 0.95 이상 · 되돌릴 수 없음 0.95 이상 → 거부
4. 나머지는 사용자 확인

자체 시험

보정 178행 · 평가 186행

일상 허용 28/33 · 위험 허용 0/68 · 편집 0/6 · 주입표시 11/11

▲ 한계 — 별 1개(09-29) · 독립 검증 없음 · 제작자도 비권장

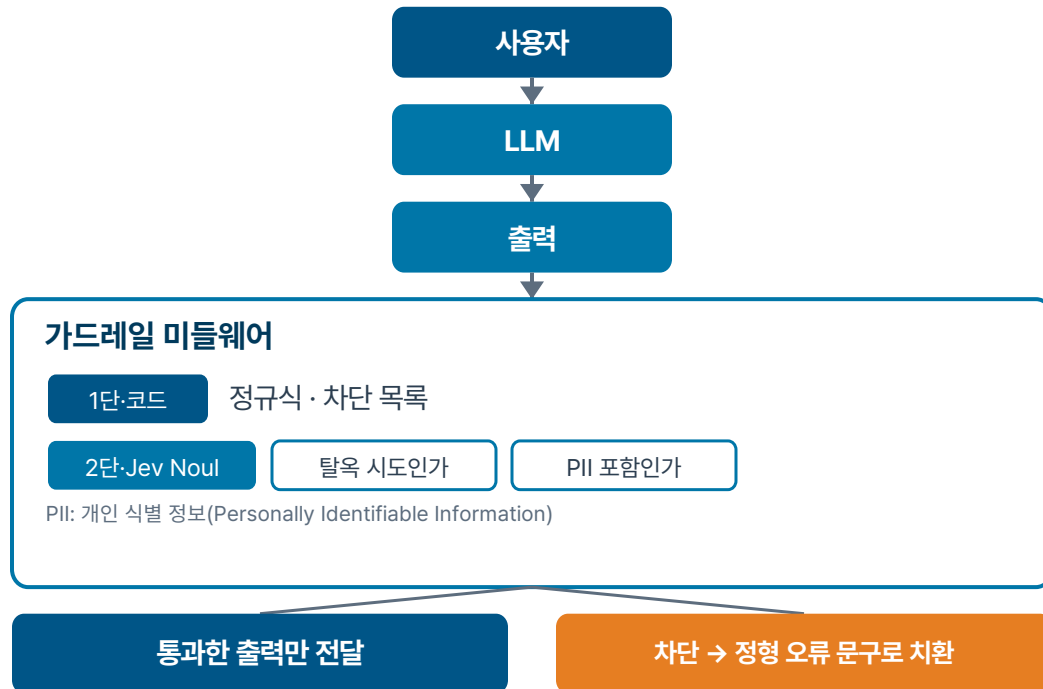
공통 교훈 — 빠른 1차 판정 + 느린 2차 판정 + 결정적 규칙 + 사람 승인의 다층 구성



KEY INSIGHT

Jev는 결정적 규칙 뒤 빠른 2차 판정에서만 신뢰 가능 — 판정모델도 탈옥 문장에 흔들리는 탓

LLM 출력 직전, 탈옥과 개인정보 노출을 가려내는 인라인 가드레일



차용할 원리 — 1차 빠른 판정 후 걸린 건만 2차 정밀 판정

설계 제약

- 적대적 내용에 흔들리는 약점 — 이 설계도의 정면 과제
- 정형 개인정보(주민번호·카드번호)는 정규식이 1차 판정
- Noul 은 confidence 없어 확률 문턱으로 분기
- 반대 명제 두 질문 금지 · 대체 문구는 템플릿
- 미확인 — '가드레일 미들웨어' 표기, 원문 명칭 아님

참고 패턴 — Claude Code 자동 모드

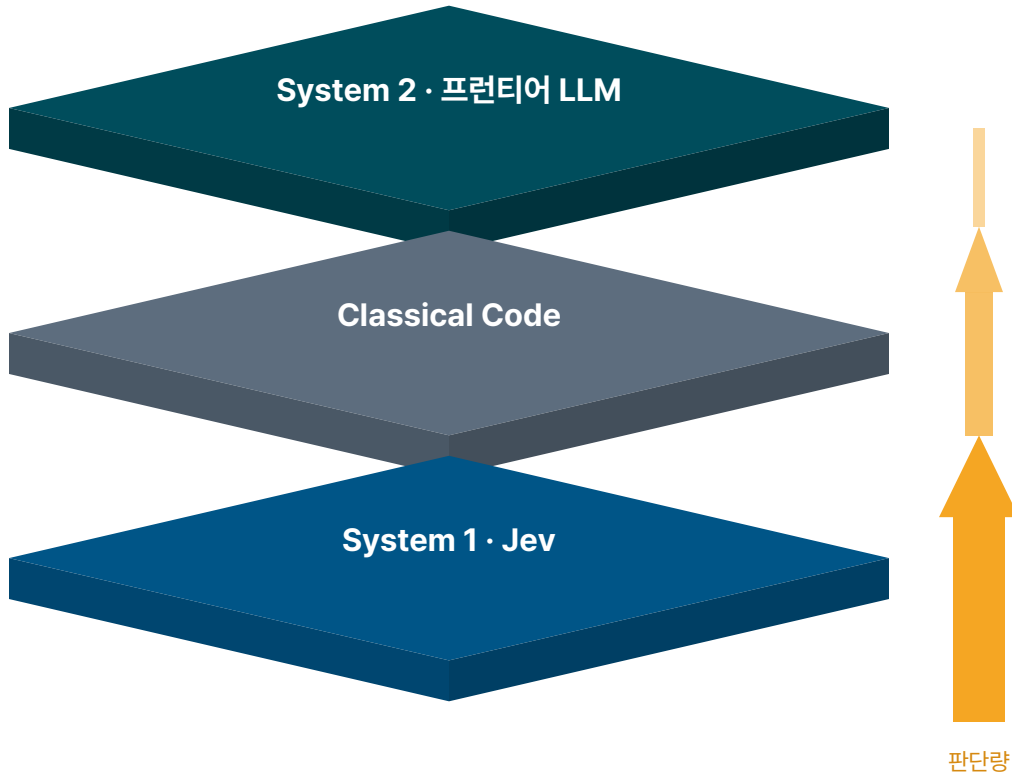
1단 빠른 판정
오탐 8.5%

2단 재판정
오탐 0.4%

전체 파이프라인 오탐 0.4% (실트래픽 기준)

분류기는 도구 출력을 보지 않아 주입 지시를 차단

판정 지연 — 추정 50~100ms / 공식 70~500ms



많은 판단이 아래층에서 끝나고 일부만 위로 올라감

원 자료 맷음말(재진술): LLM 에 선택을 맡기던 방식에서 타입 확률로 분기를 코드화하는 방식으로 전환

TypeSafe 선언문: 지능을 코드 옆에서 호출하는 기본 요소로, 감사 · 시험 가능성이 안전을 만들

KEY INSIGHT 가운데 코드 층이 Jev 약점을 메우는 자리 —
계산 · 날짜 · 차단 규칙을 코드로 끌어내리지 않으면 아래층 속도 이점이 오판으로 상쇄

층	맡는 일	받아 주는 약점
위층 System 2	깊은 추론 · 자유 기술 · 응답 문장 생성 라우팅 캐스케이드의 깊은 추론 갈래	4 단계 추론 9 생성 불가
가운데층 Classical Code	확신도 문턱 If 문 · 폴백 금액 · 개수 · 날짜 계산 정규식 · 차단 목록 Jev 약점을 메우는 자리	2 수학 · 숫자 3 날짜 비교 6 적대적 내용
아래층 System 1 · Jev	대량 · 고속 · 저비용 판단 분류 · 점수 · 예/아니오 설계도 1~6 의 판정 단계	— (약점을 위 두 층에 넘김)

SECTION 05

한계와 도입 전략

LLM과 Jev의 역할 분담 기준과 약점 영역



KEY INSIGHT

작업 성격이 도입 기준 — 새 대안은 LLM, 정해진 선택은 Jev 가 담당

Jev 가 못 하는 일

한계 01

후보 밖 선택 불가

정의되지 않은 새로운
대안은 만들지 못함

한계 02

깊은 추론 불가

여러 단계를 거치는 분석은
생성형 LLM 의 영역

한계 03

생성 작업 불가

글쓰기·요약·코드 생성은
지원 범위 밖

역할 분담 기준

오래 생각해야 하는 일과 빠르게 골라야 하는 일의 구분

LLM — 오래 생각해야 하는 일

- 창작·설계처럼 새 대안을 만드는 작업
- 여러 단계를 거치는 복잡한 분석
- 설명이 필요한 문장 생성

Jev — 빠르게 골라야 하는 일

- 정해진 선택지 중 분류·라우팅 판단
- 위험 여부 같은 예/아니오 판정
- 초당 수천 건의 반복 분기 처리



KEY INSIGHT

Jev 는 LLM 루프 사이사이의 짧은 판단을 흡수 —
긴 사고는 LLM, 반복 판단은 Jev 로 나눌 때 비용과 지연이 함께 감소

구분	LLM (Claude · GPT)	구조화 출력 LLM	Jev
출력	글 (텍스트)	형식을 강제한 텍스트	타입이 정해진 값 + 확률
만드는 방식	토큰을 하나씩 생성	토큰을 하나씩 생성	한 번에 판단을 반환
응답 시간	수 초	수 초	0.1초대
출력 비용	토큰마다 과금	토큰마다 과금	무료
확신 정도	알 수 없음 (과신)	알 수 없음	판단마다 확률 · 신뢰도
맥락 기억 · 긴 추론	가능	가능	불가
알맞은 일	글쓰기 · 코딩 · 추론	정보 추출 · 양식 채우기	분류 · 라우팅 · 점수화 · 필터

구조화 출력(JSON 모드)은 형식만 강제할 뿐 토큰을 순차 생성 — Jev 는 텍스트 생성 단계 없이 결정 자체를 반환



계산과 날짜 비교에 약하고 주입 지시에 흔들리는 Jev 1.13 약점

05 · 한계와 도입 전략

1 문자 그대로 읽기

원 발표 소개

의도 추론 없음

→ 경계 사례를 질문에 명시

2 수학 · 숫자

원 발표 소개

개수 세기 · 색상값 근접 판단 불안정

→ 계산은 코드로 처리

3 날짜 · 시간 비교

원 발표 소개

날짜를 순서 값이 아닌 텍스트로 취급

→ 비교는 코드로 처리

4 간접 추론

다단계 추론 · 이중 부정에 약함

→ 질문을 한 단계로 분해

5 관련 없는 큰 상태

무관한 내용이 늘수록 정확도 하락

→ 필요한 부분만 상태로 전달

6 적대적 내용

주입된 지시에 흔들릴 수 있음

→ 보안 판단 단독 사용 금지

7 모순된 지시

지시문과 기준 불일치 시 혼란

→ 기준 문구 일관성 점검

8 구조적 불변식 없음

반대 질문 둘의 합이 1이라는 보장 없음

→ 한쪽 질문만 사용

9 생성

원 발표 소개

텍스트 생성용 학습 없음

→ 생성은 LLM 과 조합

보강 — 원 발표는 4개(세기 · 날짜 · 문자 그대로 · 생성 불가)만 소개, 공식 문서 기준 9개로 확장

KEY INSIGHT 약점 대부분이 코드로 대신할 수 있는 영역 —

계산 · 날짜는 코드가, 적대적 입력은 결정적 차단 목록이 먼저 거르는 구성이 전제



다른 수단에 넘겨야 할 Jev 공식 약점 아홉 종

05 · 한계와 도입 전략

공식

Jev 1.13 약점 문서 9항목을 원 발표의 대체 수단 4목음으로 정리

대체 수단

해당 공식 약점



생성 모델에 맡김

9 생성 — 텍스트 생성용으로 학습되지 않음



일반 코드로 처리

2 수학 · 숫자 — 개수 세기, 색상값 근접 판단 불안정

3 날짜 · 시간 비교 — 날짜를 순서값이 아닌 텍스트로 다룸

8 구조적 불변식 없음 — 반대 명제 두 개의 확률 합이 1 이라는 보장 없음



문제와 입력을 좁히고
필요하면 LLM 사용

1 문자 그대로 읽기 — 의도 추론 없음, 경계 사례 명시 필요

4 간접 추론 — 다단계 추론 · 이중 부정에 약함

5 무관한 큰 상태 — 관련 없는 내용이 늘수록 정확도 하락

7 모순된 지시 — 지시문과 기준이 어긋나면 혼란



경계 테스트 + 코드 · 사람 제어

6 적대적 내용 — 입력을 적대적으로 보지 않아 주입 지시에 흔들림 · **중대한 권한 판단에 단독 사용 금지**



한국어 주의 (공식 모델 문서)

영어가 주력 학습 언어 · 한중일 문자는 지원하나 정확도가 같지 않음 — 한국어 업무는 운영 전 자기 데이터 시험 필수



KEY INSIGHT

없는 선택지를 만들어 내지 않는 것과 판단이 옳은 것은 별개 —

약점 영역을 입력 단계에서 걸러 내지 않으면 형식이 맞는 오답이 그대로 업무 흐름에 들어감



KEY INSIGHT

판단 난이도와 반복 빈도 두 축이 도입 범위를 가름 — 오판하면 과잉투자·품질저하로 이어지니 이 기준으로 결정

도입 여부를 가르는 두 축 — 판단 난이도와 반복 빈도

판단 난이도 ↑

난이도 높음 · 빈도 낮음 LLM + 사람 확인 드문 예외는 설명 가능한 생성형 판단과 사람 승인	난이도 높음 · 빈도 높음 하이브리드 Jev 가 1차로 가르고 애매한 건만 LLM·사람에게
난이도 낮음 · 빈도 낮음 규칙 기반 자동화 조건문만으로 충분 Jev 투입도 과함	난이도 낮음 · 빈도 높음 Jev 초당 수천 건 분기 할루시네이션 없이 저비용

반복 빈도 →

도입 체크리스트

역할 분담표만으로는 못 정하는 선택 기준

- **반복 빈도 먼저 확인**
하루 발생 건수를 로그로 확인
- **성문화 가능성 점검**
if-else 로 적히면 Jev 후보
- **예외 경로 확보**
애매한 건은 하이브리드로 전달

SECTION 06

오픈소스 대안과 생태계

JevBench 순위와 하드웨어 요구량으로 본 대체 모델 후보



KEY INSIGHT

라이선스 때문에 탈락하는 후보는 거의 없음 — 후보를 가르는 기준은 라이선스보다 독립 측정 결과

프로젝트	방식	기반 모델과 크기	라이선스	특징
Laya	인코더 학습	ModernBERT-large 421M · 다국어 322M	Apache-2.0	별 최다(1.9만~2.7만), CPU 실행
Kev	LoRA 미세조정	Qwen 0.8B · 4B · 9B · 27B	Apache-2.0	/v1/systemone 호환, SDK 그대로 사용
SemIf	로짓 읽기	Qwen3.5 4B 동결	MIT	브라우저(WebGPU)에서 실행
so1	로짓 읽기	임의 오픈 LLM (Qwen3.6-27B 로 측정)	Apache-2.0	한 번의 순전파에 여러 질문 처리
Jeff	전체 미세조정	Qwen3.5 0.8B · 2B	코드 MIT 가중치 Apache-2.0	DGX Spark 로 2~3.5시간 학습
Cygnnet	로짓 읽기	Gemma 4 12B 동결	코드 MIT 가중치 Apache-2.0	Gemma 금지 용도 정책 적용
Winnow-12B	미세조정	Gemma 4 12B	미확인 (Gemma 정책 적용)	대화와 이미지 입력 유지, 16GB GPU
decider · JevK5 · Plumb	미세조정	Qwen3.5-4B 계열	Apache-2.0(decider)	JevBench 종합 4~7위
Ollaya	실행기	위 모델들을 내려받아 실행	Apache-2.0(런타임)	로컬 /v1/systemone 제공

Source: 각 프로젝트 저장소 · github.com/blockbrain-ai/cygnnet-recipe · jevmodel.org/jev-vs-winnnow · benchmarkheaven.com/jev-models



오픈 모델 두 개가 Jev 를 넘어선 JevBench v1.5.4 종합 순위

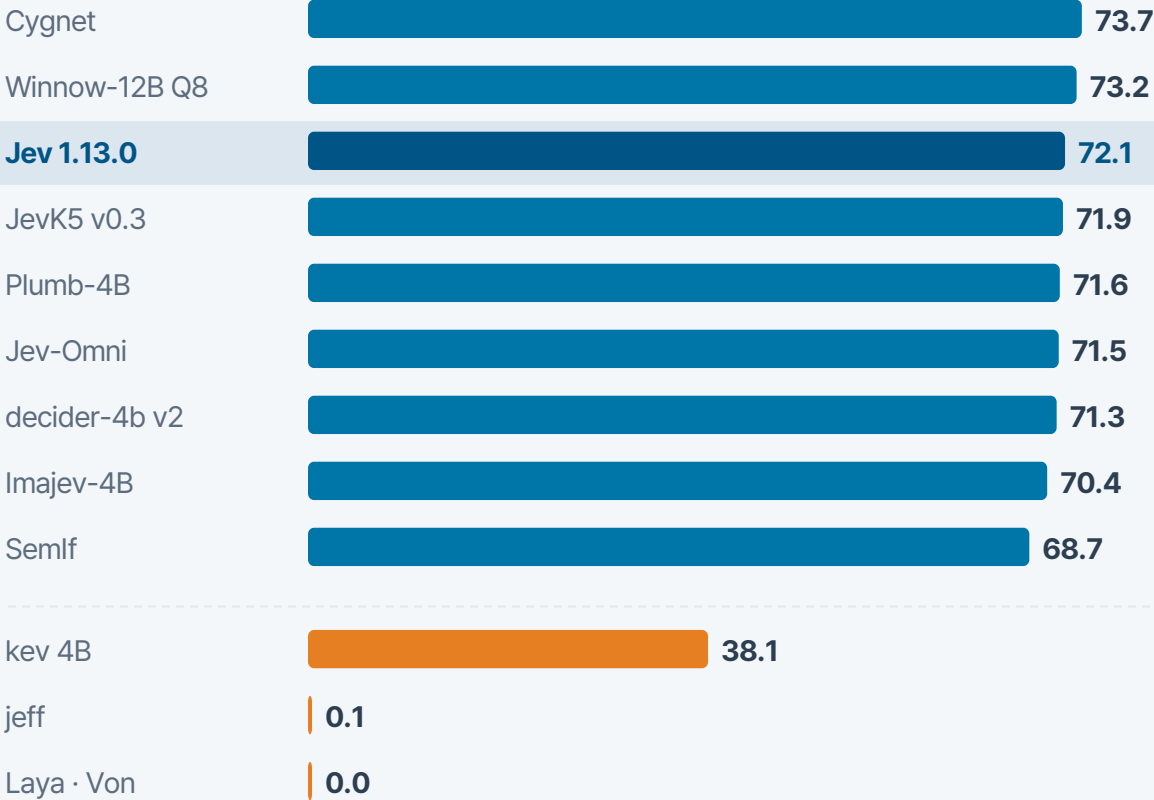
06 · 오픈소스 대안과 생태계

KEY INSIGHT

1위와 Jev 의 차이는 1.6점 —
판 버전이 바뀔 때마다 순위가 뒤집힌 범위여서 인용할 때 버전 표기가 필수

종합 점수 (100점 만점)

순위 1~9위 · 12위 · 하위권



순위를 읽을 때 볼 점

- 1위와 Jev 의 차이 1.6점,
4위 JevK5 와 Jev 의 차이 0.2점
- 하위 세 개의 0점대 점수는 판단력 50 미만
감점의 결과 — 인코더 계열이 여기에 해당

판 버전에 따라 달라진 1위

버전	1위	Jev
v1.4.2	decider-4b v2 64.1	63.3
v1.4.2.2	Imajev-4B 67.37	63.29 (4위)
v1.5.4	Cygnnet 73.7	72.1 (3위)

Winnow-12B 는 v1.4.2.2 에서 8위(55.6),
v1.5.4 에서 2위 — 같은 모델, 다른 채점 규칙

KEY INSIGHT

속도와 비용 측은 실측이 아닌 가정 —

종합 순위 하나보다 판단력과 보정 측을 따로 읽어야 오판을 피함

판단력

실측

단계별 우연 보정 정확도
절반은 비공개 문항으로 채점

보정

실측

ECE(기대 보정 오차)와 정답 분포 충실도
라벨만 내는 시스템은 0점

속도

가정

p50 · p95 지연의 로그 점수
자체 호스팅은 지연 2배 + 0.15초로 계산

비용

가정

1천 판단당 달러의 로그 점수
자체 호스팅은 호스팅 요금으로 추정

종합 점수 계산

$$\text{종합} = 4 \div (1/\text{판단력} + 1/\text{보정} + 1/\text{속도} + 1/\text{비용})$$

- 네 측의 조화평균 — 한 측이 낮으면 종합이 크게 하락
- 판단력이 50 미만이면 $(\text{판단력} \div 50)^2$ 을 곱해 감점
- 공개와 비공개 점수 차가 크면 판단력 감점
- 시스템당 1,624 판단 = 공개 904 + 비공개 720

저자가 밝힌 한계

영어 전용

한국어 문항 없음

짧은 수작업 사례

긴 입력 · 실제 업무 문항과 차이

개인 운영

판 버전마다 채점 규칙 변경

비공개 문항도 전송

예측을 받으려면 시스템에 보내야 함

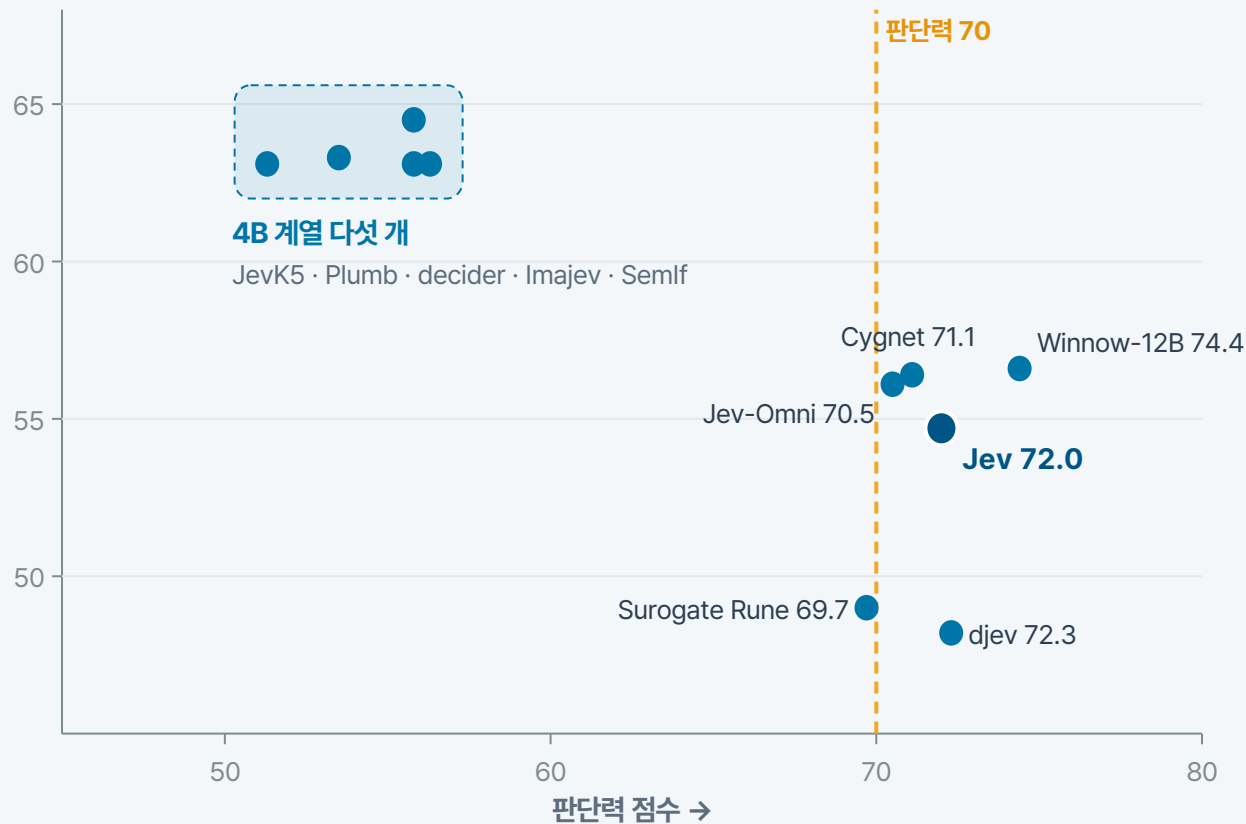


KEY INSIGHT

4B 계열은 속도와 비용 점수로 종합을 메움 —

어려운 판단을 맡길 자리에서는 종합 순위가 후보 선정 기준으로 부적합

비용 점수 ↑ (높을수록 저렴)



두 무리로 갈린 결과

판단력 70 이상은 다섯 개

Winnow-12B · djev · Jev · Cygnet · Jev-Omni
모두 12B 이상이거나 확산 모델

4B 계열은 판단력 51~56

비용 점수 63~64.5 (1천 판단당 \$0.015~0.017)
Jev 는 비용 점수 54.7 (\$0.032)

- 비용 \$0.053 때문에 종합 20위에 머문 djev, 판단력 점수는 72.3
- 비공개 문항 정답률(v1.4.2): Jev 36.7%, decider-4b v2 34.7%, Winnow 33.1%

처음 보는 문항에서는 원본이 여전히 앞섬



KEY INSIGHT

독립 검증에서 Laya 외 전부 Jev 우위 — 자기 보고 수치는 후보 선별용일 뿐 채택 근거로는 부족

프로젝트	자기 보고 수치	같은 조건의 Jev	독립 검증 결과
Laya (미세조정)	typed-decisions 0.766 ECE 0.081 · 32.8ms(T4)	0.727 · ECE 0.246 236~276ms	반박됨 — 자기 데이터로 측정 미세조정 전 0.362, Banking77 0.425→0.870
Kev-27B	정확도 0.851 · Brier 0.225 자동화 가능 비율 0.69	0.857 · 0.211 0.70	반박됨 — JevBench kev 4B 38.1 생태계 조사가 반박으로 분류
so1 + Qwen3.6-27B	정확도 73.7% · ECE 0.020 지연 582ms	72.7% · 0.144 지연 710ms	독립 검증 없음 저자 측정만 존재
Semlf 4B	Jev 와 일치율 0.845 JSON 생성 대비 5.21배 빠름	기준 0.883	지지됨 — JevBench 68.7(12위) 판단력 51.3 · 보정 84.0
Jeff 2B	종합 83.1% · 분류 96% 지연 24ms	분류 77.0% · BBH 94.3%	재현되지 않음 — JevBench 0.1 같은 이름의 다른 구현일 가능성
NanoJev	ViZDoom 128/128 게임 제어 전용	56/128	자기 보고만 존재 다른 과제로 일반화할 근거 없음

생태계 조사 판정 — 복제 모델 10개 가운데 주장 5건은 독립 근거로 반박, 3건은 제한적으로 지지

Source: OMDM 조사 페이지 2절 (5) · github.com/cl-victor1/jev-ecosystem-survey · benchmarkheaven.com/jev-models



보정된 확률과 입력 한도에서 차이가 벌어진 오픈 후보의 기능 범위

KEY INSIGHT

보기 선택 · 척도 · 예 확률 세 형식과 API 호환은 거의 충족 —
실제 교체를 막는 항목은 보정된 확률과 긴 입력

Jev 기능	Jev 1.13.0	Kev	4B 미세조정 계열	Semlf · so1	Cygnnet	Laya
Choice (보기 선택)	최대 255개	지원	지원	보기 라벨이 한 토큰	최대 255개	20개 넘으면 급락
Score · Noul	지원	지원	지원	지원 (척도는 so1)	지원	지원, 척도가 높임
한 요청에 여러 질문	지원, 지연 거의 불변	지원	지원	뭉으면 6~9% 흔들림	미확인	배치 지원
보정된 확률	보정 88.0	ECE 0.041 (자기 측정)	보정 85~88	84.0 · 약 5%p 과신	보정 87.0	기본 ECE 0.466
입력 한도	64k	소형 384토큰 학습	미확인	기본 모델에 따름	16k	512~1,024 토큰
이미지 입력	불가	불가	Imajev 는 가능	비전 래퍼로 가능	미확인	불가
내 데이터 미세조정	불가	가능	가능	불필요	불필요 (동결)	사실상 필수
API 호환	원본	호환	Ollaya 경유	자체 API	호환	Ollaya 경유
지연 (p50)	0.62초	18ms (H100, 4B)	0.18~0.23초	0.23초	0.23초	30~40ms

API 호환: /v1/systemone 요청 형식을 그대로 받는지 여부 · 4B 미세조정 계열: JevK5 · Plumb · decider · Imajev



1GB 미만부터 48GB 급까지 벌어지는 후보 모델의 GPU 메모리 요구량

06 · 오픈소스 대안과 생태계

KEY INSIGHT

4B 계열은 3~5GB 로 일반 GPU 한 장에 적재 —
장비 한 대에 후보 여러 개를 함께 올려 같은 문항으로 비교 가능

모델	크기	양자화	GPU 메모리	지연	측정 장비
Laya	421M	Q8	1GB 미만	32.8ms	Tesla T4
Kev-0.8B	0.8B	bf16	미표기	47ms · 한국어 30 기준 52ms	Mac MLX · M3 Max
decider-4b	4B	Q4_K_M	2.71GB	약 22ms	추정치
JevK5	4B	Q8	4.48GB	약 13ms	H100
Kev-4B · Kev-9B	4B · 9B	미표기	미표기	한국어 30 기준 307ms · 539ms	M3 Max 36GB
Winnow-12B	12B	Q8	12.67GB	미표기 (16GB GPU 권장)	RTX 5070 Ti
Cygnnet	12B	원본 정밀도	48GB 급에서 시험	50~66ms · 동시 8 요청 시 초당 38.5건	RTX A6000 · L40S

설치 예

```
pip install laya
```

```
uv run python -m kev.serve --run jaredpalmer/kev-0.8b --port 8009
```

■ 5GB 이하 ■ 16GB 이하
■ 48GB 급



KEY INSIGHT 판정은 Jev, 생성은 오픈소스 병행 — 속도·정확도·비용·GPU 네 지표 비교 결과

평가 기준	Jev 공식 모델 판정 전용 · TypeSafe AI	오픈소스 대안 범용 생성형 LLM 전반
처리 속도	● 초당 수천 건 판정 처리	○ 요청 1건에 수백 ms~수 초
정확도 · 환각	● 할루시네이션 사실상 없음	○ 모델 · 프롬프트별 편차 존재
비용 구조	● LLM 호출 대비 수백 배 저렴	○ 자체 GPU 인프라 상시 비용
GPU 요구량	● API 호출 — 자체 GPU 불필요	○ A100급 이상 상시 점유 필요

선택 기준

분기 판정 · 라우팅처럼 속도와 비용이 걸린 호출은 Jev
장문 생성 · 창작 · 설명이 필요한 작업은 오픈소스 대안을 병행

THANK YOU

감사합니다

판단형 AI 모델 Jev 입문

MSAP.ai

msap.ai

 <https://www.msap.ai/>

 02-6953-5427

 hello@msap.ai