

ko-pii — 한국어 개인정보 검출·가명화 라이브러리

룰·사전·체크섬만으로 동작하는, 설명가능하고 빠른 한국어 PII 엔진.

`pip install ko-pii` · MIT · github.com/Marker-Inc-Korea/ko-pii · v1.14.0

1. 한눈에

항목	값
무엇	한국어 문서의 개인정보(PII) 검출 + 가역 가명화 Python 라이브러리
방식	규칙 + 사전 + 체크섬 — ML 없이, 외부 의존성 0 (코어)
대상	33종 한국 PII (주민·여권·카드·사업자·법인·운전면허 + 이름·주소·기관 등)
정확도	결정적 ID(주민·카드·여권) F1 $\approx$ 1.000 · 외부 벤치 KDPII 0.660
속도	0.19 ms/문서, ~5,350 문서/초 (CPU 1코어, GPU 불필요)
비용	100만 문서 $\approx$ 3분, \$0
품질	792 테스트 · CI(Python 3.10~3.13) · 법적근거 매핑 · 가역 Vault

한 줄: 한국어 + 온프레임 + 컴플라이언스 + 구조적 PII 라는 틈새에서, 설명가능하고 즉시·무료로 동작하는 실전 도구.

2. 왜 만들었나 — 푸는 문제

개인정보보호법은 PII 마스킹/가명화를 요구하지만, 한국어 환경엔 마땅한 도구가 없다:

- **해외 도구(Microsoft Presidio 등)** 는 한국 특화 엔티티(주민번호·사업자·학과·직책·법정동)를 사실상 못 잡는다 (KDPII F1 0.27).
- **LLM 모델** 은 대화체엔 강하지만 느리고(수백 ms~초), 비싸며, 결정적 ID(주민번호)를 체크섬 검증 없이 환각·누락한다.
- 공공·금융·의료의 **대량 배치 마스킹** 은 비용·속도·온프레임·감사가능성이 핵심인데 위 둘 다 부적합.

ko-pii는 이 공백 — *한국 구조적 PII를 빠르고·정확하게·설명가능하게* — 를 메운다.

3. 무엇을 하나 — 기능

검출 (Detection) — 33종, 세 가지 메커니즘

- 체크섬 — 주민·카드(Luhn)·사업자·법인 검산 → 결정적 ID 정밀도 ≈ 1.000
- 사전 — 성씨(187)·직책·기관·대학·학과·행정구역·법정동(1만) 매핑
- 한국어 문맥 점수 — 이름처럼 비결정적인 PII를 주변 신호로 판정

각 검출은 왜 잡혔는지(`evidence` — 발동 신호) · 위험등급(`risk_level`) · 법적 근거(`legal_basis` — 해당 개보법 조항)를 함께 돌려준다 → "PII다"만 출력하는 ML과 달리 검출 근거를 사람이 검토·감사할 수 있다. 감사·컴플라이언스 보고에 직결된다.

가명화 (Anonymization) — 5 모드 × 6 전략

- 모드(차단 강도): PARANOID / STRICT / BALANCED / PERMISSIVE / AUDIT
- 전략(치환 방식): tokenize / redact / asterisk / partial / hashed / fpe
- 가역·Vault(AES-GCM 암호화 + 감사로그)로 `reveal()` — 키까지 쪼개 보워

4. 어떻게 검출하나 — PII 종류별 방식

검출 방식은 PII 종류마다 다르다. 점수 채점 산식은 이름에만 쓰고, 나머지는 수학적 검산·형식 매칭이다.

(a) 결정적 PII — 체크섬 검산 (산식 아님)

주민·카드·여권·사업자·법인. 정규식으로 후보를 잡고 체크섬을 계산해 검산한다.

검산 통과 → confidence 1.0, 형식만 맞으면 낮은 confidence(예 0.7). 가짜는 체크섬이 걸러내 오탐 ≈ 0 .

(b) 패턴 PII — 정규식 형식 매칭

전화·이메일·IP·URL. 한국 번호 체계·이메일 문법 등 형식 규칙에 맞으면 검출 (confidence 1.0).

(c) 이름(PERSON) — 가산 점수 산식 ← 유일하게 "점수 채점"하는 검출

이름은 다정한 스 어오므로 주변 시황을 점수로 더해 인계와 비교한다

5. 성능 (실측)

정확도 — KDPII 전체 4,891문서 (인간 라벨, 대화체), 동일 채점

시스템	F1
ko-pii (룰)	0.660
Gemma-4-31B (LLM, 참고선)	0.796
Gemma-4-E4B (최소 Gemma4, ~4B)	0.613
Presidio (한국어 적응)	0.273
openai/privacy-filter (660M ML)	0.264

- ko-pii는 룰만으로 31B LLM(Gemma)의 83% 정확도에 근접 — 단 LLM은 ~190배 느리고 GPU가 필요하다(아래 속도). 룰·NER 도구(Presidio·openai/PF)는 큰 폭으로 앞선다.

6. 경쟁 비교 — 강·약점

비교 대상 — 해외 룰/NER: Microsoft **Presidio**(정규식 룰 + spaCy 한국어 NER) · **openai/privacy-filter**(660M 트랜스포머 NER). LLM: **Gemma-4-31B**(측정) 등 GPT·Claude 류 (프롬프트로 PII 추출).

측	ko-pii	Presidio·openai/PF (룰·NER)	Gemma 등 (LLM)
한국 결정적 ID	✅ 체크섬 ≈1.0	❌ 주민번호조차 0	❌ 검증 없음(오탐)
한국 특화 엔티티	✅ 학과·직책·법정 동	❌ 라벨 없음	△
속도·비용	✅ 무료·즉시	○	❌ 느림·고비용
온프레임(데이터 안 나감)	✅	✅	△

7. 설계 철학 — "전부 마스킹"이 아니라 선택 가능한 트레이드오프

ko-pii는 프라이버시 ↔ 정밀도/효용을 사용자가 모드·전략으로 고르는 스펙트럼으로 본다. 한계가 아니라 컴플라이언스 현장의 요구를 반영한 의도된 설계다:

- **모드:** STRICT (운영 권장, MEDIUM+ 차단)부터 BALANCED (FP 최소화, 사업장 대표번호 같은 MEDIUM은 통과) PARANOID (전부 차단)까지 — 도메인별 선택.
- **partial 전략:** 880101-1***** 처럼 생년+성별만 노출 — 공문서 표준 부분마스킹 규약 호환.
- **fpe 전략:** 자릿수·구분자·체크섬 형태를 보존해 분석·통계 호환 (개보위 「가명정보 처리 가이드라인」 권장 기법).

완전 비식별이 필요하면 tokenize / redact + STRICT / PARANOID . 효용·양식 보존이 필요하면 partial / fpe . 하나의 정답이 아니라, 현장이 고르는 스펙트럼.

8. 엔지니어링 품질

- 792 테스트 · CI(Python 3.10~3.13) · CHANGELOG · PyPI 정식 배포
- 법적근거 매핑: 검출마다 개보법 조항 + 근거 등급
- 가역 Vault: AES-GCM + 키 유도(KDF) 지문 + 감사로그 — 컴플라이언스급
- 공개 벤치마크 + 영문 문서 — 제3자 재현 가능
- 적대적 보안 검증: 다중 에이전트 누출 스윕으로 PII 평문 누출 13건을 찾아 즉시 수정(v1.12.0). 검증 체계가 작동한다는 성숙도 신호.

9. 정직한 한계 (숨기지 않는다)

- 자유 대화체 재현율이 낮다 — 맨 이름·구어 주소는 룰의 보수적 설계상 약함 (KDPII PERSON 0.135, ADDRESS 0.241). 챗/SNS처럼 "하나도 놓치면 안 됨"이면 ML 하이브리드가 필요하다. ko-pii는 이걸 *정직하게 인정하고* [classifier] 하이브리드를 제공한다.
- 룰 기반 천장 — 학습으로 새 맥락에 일반화하는 ML과 달리, 신규 패턴은 수작업·사전 갱신이 필요하다.

10. 결론 — 가치 한 줄

**"가장 정확한 PII 검출기"가 아니라, 한국어 PII에서 비용·속도·결정성·온프레임·설명가능성의 최적 균형점".*

대량·온프레임·컴플라이언스·구조적 PII 시나리오에서 — 해외 도구를 정확도로 앞서고, 둘 다 못 가진 *체크섬 결정성 + 법적근거 + 빠른* 속도를 제공한다. 자유 대화체라는 알려진 천장은 ML 하이브리드로 정직하게 보완하는, 좁지만 깊은 실전 가치를 가진 라이브러리.

측정: KDPII v1.1 test(4,891문서, 인간 라벨) · 2026-06