

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 · v1.13.0

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	772 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()` — 키 가진 쪽만 복원
- **구조 보존**: CSV/XLSX는 표 구조 유지하며 컬럼 단위 처리

입력

- 텍스트 + 파일 파서: **HWP·PDF·DOCX·XLSX·CSV** (`pip install ko-pii[file]`)

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 검산**한다.

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

(b) 패턴 PII — 정규식 형식 매칭

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

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

이름은 단정할 수 없으므로 **주변 신호를 점수로 더해 임계와 비교**한다.

점수 = 0 에서 시작 → 신호를 가산:

종류	신호	점수
문맥(주변)	양식 라벨 성명: 직전	+0.50
	직책 사무관 · 대표 인접	+0.35
	조사 이/가/은/는 붙음	+0.35
	전화·주민·이메일 25자 내	+0.40 (2자면 +0.20)
	같은 문장 기관명	+0.10
	성씨로 시작	+0.15
이름 자체	이름끝 음절 · 음절 통계 · 나이/성별 인접	+α
감점	일반어 -0.40 · 1글자 -0.30 · 영문/숫자 -0.20	

판정: 합 ≥ 0.50 → PERSON (점수가 곧 confidence). 짧고 약한 이름은 0.60 필요.

이 가산 점수 산식은 이름 검출에만 적용된다 — (a)(b)는 점수 없이 검산/형식이다.

실측 예시:

입력	누적	결과
성명: 홍길동	성씨 + 필드라벨 + 이름신호	0.95 ✓
장혁이 떠났다	성씨 + 조사 + 이름신호	0.80 ✓
김철수 010-1234-5678	성씨 + 전화인접 + 이름신호	0.85 ✓
홍길동 왔다	성씨 + 이름신호만 (맥락 0)	~0.45 거부 ✗

맥락(조사·직함·양식·인접 PII)이 있어야 검출 = 오탐 방지. 맨 이름은 일부러 보수적으로 흘려보낸다.
조사·직함·양식 라벨 = 해외 도구엔 없는 한국어 특화 신호.

최종 마스킹 결정 — 모드가 임계를 바꾼다

검출된 PII의 confidence를 **모드의 confidence_min** 과 비교해 BLOCK (마스킹) /

REVIEW (검토 권고) / PASS (통과)를 결정한다. **모드를 바꾸면 마스킹 컷오프가 바뀐다:**

모드	마스킹 임계 (confidence_min)
PARANOID	0.5
STRICT	0.7
BALANCED	0.8
PERMISSIVE	0.95

예: confidence 0.55짜리 약한 이름은 PARANOID(0.5)에선 마스킹되지만 STRICT(0.7)·BALANCED(0.8)에선 통과한다. (검출 임계 0.50 자체는 하드코딩 — 직접 노출하려면 작은 수정으로 파라미터화 가능.)

5. 성능 (실측)

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

시스템	F1
ko-pii (롤)	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 롤만으로 31B LLM(Gemma)의 83% 정확도에 근접 — 단 LLM은 ~190배 느리고 GPU가 필요하다(아래 속도). 롤·NER 도구(Presidio·openai/PF)는 큰 폭으로 앞선다.
- 크기 효과 — 대화체 KDPII에선 작은 E4B(0.613)가 오히려 ko-pii(0.660) 미만, 큰 31B(0.796)만 롤을 능가. 행정체 생성셋에선 반대(E4B 0.882 > ko-pii 0.790) — 롤의 강점 도메인이 분명.
- 결정적 ID(주민·이메일·IP·외국인등록·전화·여권·차량): per-label F1 0.91~1.00
- 공정 비교 — 위 전체 점수는 해외 도구가 라벨 자체가 없는 항목(AGE·POSITION 등)에서 0점이라 격차가 커진다. 각 도구가 실제 지원하는 카테고리만으로 좁혀도 ko-pii가 앞선다: vs openai/PF 0.61 : 0.37 (그쪽 7개 라벨), vs Presidio 0.87 : 0.65 (그쪽 9개 라벨).

정확도 — 생성 평가셋 540문서 (행정/서식체, 검증 gold), 동일 채점

KDPII가 대화체라 LLM에 유리한 것과 대비되도록, 행정/서식체 합성 문서셋도 별도 측정.

시스템	F1	P	R
Gemma-4-31B (LLM)	0.964	0.963	0.966
Gemma-4-E4B (최소 Gemma4, ~4B)	0.882	0.925	0.843
ko-pii (롤)	0.790	0.794	0.787
Presidio (한국어 적응)	0.483	0.794	0.347
openai/privacy-filter (660M ML)	0.451	0.445	0.457

- **gold 검증** — 540문서·3,635 span을 2단계(자동 형식검증 + LLM 적대적 감사)로 검증해 **98.8% 정확**(37건 수정). 데이터·절차 공개(`data/generated_eval.jsonl`).
- 이 셋은 소프트 속성(직책·학력 등)이 많고 gold가 LLM 생성이라 **LLM에 유리한 편향**이 있다 — 그럼에도 ko-pii의 결정적 ID는 천장급(EMAIL 0.998·전화 0.989·카드 0.988·주민 0.955).
- **작은 LLM도 강하다** — 최소 Gemma4(E4B, ~4B)가 31B의 **91% F1**(0.882)을 ~8배 작은 크기로 달성. 단 LLM은 GPU·비용·비결정성을 동반하며, ko-pii는 CPU·\$0·결정성으로 다른 운영점.

속도 (100만 문서 기준)

시스템	ms/문서	처리량	100만 문서
ko-pii	0.19	~5,350/초 (CPU)	~3분, \$0
Presidio	4.2	~238/초	~70분
openai/PF	481 (CPU)	~2/초	GPU 필요
Gemma-4-31B (LLM)	~36 (GPU)	~28/초	GPU 1장 점유

ko-pii는 31B LLM보다 정확도는 약간 낮지만(0.660 vs 0.796) **속도는 ~190배 빠르고 GPU·비용이 없다** — 대량·온프레미션 이 균형이 결정적이다.

채점: 모든 시스템을 같은 KDPII 문서·같은 매칭 기준으로 비교(ko-pii·Presidio· openai/PF는 아래 명령으로 재현, Gemma는 동일 매칭으로 별도 측정). 재현: `python -m ko_pii.eval.model_comparison data/kdpaii/test.json --mode kdpaii --include-presidio`

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·법정동	❌ 라벨 없음	△
속도·비용	✅ 무료·즉시	○	❌ 느림·고비용
온프레임(데이터 안 나감)	✅	✅	△
설명가능·법적근거	✅ evidence+조항	❌	❌
자유 대화체 재현율	△ (약점)	△	✅ 강점

→ **대체가 아니라 보완**: 구조적·대량·온프레임은 ko-pii, 대화체 고재현율은 LLM 하이브리드 (ko-pii[classifier] 로 룰+ML 결합 제공).

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

ko-pii는 프라이버시 ↔ 정밀도/효율을 **사용자가 모드·전략으로 고르는 스펙트럼**으로 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 . 효율·양식 보존이 필요하면 partial / fpe . **하나의 정답이 아니라, 현장이 고르는 스펙트럼.**

8. 엔지니어링 품질

- **772 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 속도*를 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