

ko-pii

한국어 PII 검출·가명화 라이브러리 — 구현 & 사용법

규칙 + 사전 + 체크섬 · ML 없이 · 외부 의존성 0 · 33종 PII

```
pip install ko-pii
```

빠른 시작

```
# ① 검출 – DetectionResult 리스트 반환
from ko_pii.detect import detect_all
```

```
for d in detect_all("홍길동 사무관 010-1234-5678, 900101-1234567"):
    print(d.label, d.text, d.start, d.end, d.confidence)
# PERSON 홍길동 0 3 1.0
# PHONE 010-1234-5678 ...
# RRN 900101-1234567 ...
```

```
# ② 가명화 (가역)
```

```
from ko_pii import Anonymizer, ProcessingMode
```

```
r = Anonymizer(mode=ProcessingMode.STRICT, strategy="tokenize").process(text)
r.text          # "<PERSON_1> 사무관 <PHONE_1>, <RRN_1>"
r.vault.reveal("<PERSON_1>")  # "홍길동" ← vault 가진 쪽만 복원
```

검출 — 3가지 방식 (ML 없이)

방식	구현	대상
체크섬 <code>checksum/</code>	주민·사업자·법인 = mod-11 가중합 / 카드 = Luhn	결정적 PII → $\approx 100\%$
사전 <code>dictionaries/</code>	성씨·직책·기관·대학·학과·행정구역·법정동	인명·주소·기관
컨텍스트 점수 <code>context/</code>	가산 점수기 (아래)	이름 등 비결정적

→ `detect_all()` = 28개 검출기 순차 실행 → 33종 PII (이름 점수는 다음 장 ↓)

이름 검출 — 한국어 문맥 점수 ★

주민번호처럼 결정적이지 않은 이름은 → 주변 한국어 신호를 점수로 합산, 합 ≥ 0.50 이면 PERSON

신호	무엇인가 (한국어 단서)	점수
필드라벨	공문서·양식의 이름칸 라벨 — 성명: · 신청인 바로 뒤	+0.50
직책	직함 — 공무원(사무관 · 과장) / 민간(대표 · 부장) 인접	+0.35
조사	한국어 조사 이/가/은/는/을/를 가 붙음 (장혁이)	+0.35
결정적 PII	전화·주민번호·이메일이 25자 내 인접	+0.40
성씨	한국 성씨로 시작 (187개)	+0.15
감점	일반어 -0.40 · 1글자 -0.30 · 영문/숫자 -0.20	

있을 때 vs 없을 때 (실측): 성명: 홍길동 ✓ / 홍길동 ✕ · 장혁이 떠났다 ✓ / 장혁 떠났다 ✕

→ 조사·직함·양식 라벨 = 한국어 특화 신호 — 정규식·해외 도구엔 없는 차별점

아키텍처 — 파이프라인

입력 텍스트

- ① 정규화 `core/unicode_norm` (NFKC·제로폭 제거) + `io_/text_normalizer` (공백·줄바꿈)
↳ `offset` 역매핑 → 마스킹 위치 보존
- ② 검출 `detect_all` → 28개 검출기 (`patterns/`)
- ③ 충돌해소 정렬(시작·위험도·길이·확신도) 후 `sweep` → 겹침 제거



`DetectionResult[]` (`label·start·end·text·confidence·risk·legal_basis`)



- ④ 정책판정 모드별 BLOCK / REVIEW / PASS



- ⑤ 가명화 전략 적용 → 결과 텍스트 + Vault(복원용)

검출과 가명화가 **분리** → `DetectionResult` 만 받아 다른 파이프라인에 끼우기 쉬움

사용법 — 옵션 & CLI

```
# 검출 필터
detect_all(text, include=["RRN", "PHONE"])      # 특정 라벨만
detect_all(text, exclude=["PERSON"])

# 모드(차단강도) × 전략(치환방식)
Anonymizer(mode=ProcessingMode.PARANOID,        # 5종: PARANOID/STRICT/BALANCED/PERMISSIVE/AUDIT
            strategy="fpe")                    # 6종: tokenize/redact/asterisk/hashed/partial/fpe

# 문서 파일 입력 (파서 자동)
from ko_pii.io_ import read_text
text = read_text("민원.hwp")                  # HWP·PDF·DOCX·XLSX
```

```
# CLI
ko-pii doc.pdf -m STRICT -s tokenize -o masked.txt
ko-pii ./docs --batch --recursive --workers 4  # 디렉토리 일괄
```

구현 구조 (모듈 맵)

모듈	역할
patterns/ (26)	카테고리별 검출기 — regex + 체크섬/사전/컨텍스트 조합
checksum/	주민·카드(Luhn)·사업자·법인 검산
context/	이름 가산 점수기 + 17개 거부률
dictionaries/	성씨·직책·기관·대학·학과·행정구역·법정동(1만)
modes/	6가지 가명화 전략
vault/	가역 토큰 매핑 + AES-GCM 암호화 + 감사로그
io_/	HWP·PDF·DOCX·XLSX·CSV 파서 + dispatcher
legal/ · analytics/	법적근거 매핑 · 결합위험도·k-익명성

확장 (전부 선택 설치, 코어 불변)

```
# 가역 가명화 + 영속 Vault + 감사로그
r.vault.save("vault.json"); ReversibleVault.load("vault.json")

# 룰 + ML 하이브리드 (pip install ko-pii[classifier])
from ko_pii.classifier import HybridAnonymizer, HybridMode
HybridAnonymizer(rule, clf, mode=HybridMode.REVIEW_FLAG, classifier_threshold=0.5)
```

- **우회 차단:** 전각(0 1 0 → 010) ·제로폭 문자 정규화 후 검출 (offset 복원)
- **통합:** Presidio recognizer · MCP 서버 ([presidio] / [mcp])

성능 ① 검출 정확도 (F1)

평가셋	구분	ko-pii	openai/PF	Presidio
행정문서 (합성 PII)	자체	0.901	0.480	0.542
KDPPII (일상 대화)	외부	0.660	0.264	0.273
KLUE (신문 인명)	외부	0.419	0.155	0.000

- 결정적 PII(주민·카드·여권·계좌)는 체크섬 검증 → F1 ≈ 1.000
- 외부 공개 데이터(KDPPII·KLUE)에서도 룰·NER 도구보다 높음
- 참고선 (LLM 상한) — 자체호스팅 Gemma-4-31B KDPPII 0.796: ko-pii는 룰만으로(GPU·LLM 없이) 그 83% 도달. 대화체(LLM 유리한 셋)에서도 근접 — 구조적 문서에선 대등 이상 예상

자체 = 합성 PII로 구축 / 외부 = 인간 라벨 공개 데이터 · KDPPII·Gemma 모두 단일 매처로 전체 4,891문서 측정

성능 ② 속도 / 처리량

	처리 시간	처리량 (1코어)
ko-pii (규칙)	0.56 ms / 문서	~1,800 문서/초
신경망 모델 (RoBERTa)	42.6 ms / 문서	~23 문서/초

- 모델 로딩·GPU 불필요 → 메모리 가볍고 콜드스타트 없음
- 대량 처리: ko-pii ./docs --batch --workers N (병렬)

※ 같은 CPU. 신경망 쪽은 예/아니오 분류 기준이라 정확히 같은 작업은 아님

정리

- **33종 한국 PII** 검출 + 가명화 — 룰+사전+체크섬, **ML 없이** · 외부 의존성 0
- **정확도**: 결정적 PII $\approx 100\%$, 외부 벤치도 해외 도구 우위
- **속도**: 0.56 ms/문서 (~1,800/초), GPU 불필요
- **품질**: 738 테스트 · CI(3.10~3.13) · PyPI 배포 · 웹 데모

```
pip install ko-pii          # 코어 (외부 의존성 0)
pip install ko-pii[file]    # + HWP/PDF 파서
```

github.com/modak000/ko-pii

부록 — 이름 점수기 전체 (Q&A)

본문 슬라이드는 대표 신호만 단순화한 것. 실제 점수기는 **13개 신호 + 다단계 패스**:

분류	신호 (가중치)
문맥	필드라벨 +0.50 · 직책 +0.35 · 조사 +0.35 · 결정적PII +0.40/+0.20 · 성씨 +0.15 · 기관 +0.10
이름 자체	토큰내직책 +0.35 · 이름끝음절 +0.10 · 음절통계(가변) · 나이/성별 +0.30
감점	일반어 -0.40 · 1글자 -0.30 · 영문/숫자 -0.20
누적	같은 문서 재등장 이름 사전 부스트 (가변)

파이프라인

- **Pass 0** — 매크로 기관+이름+직책 정규식 → 0.95 즉시 확정
- **Pass A** — 후보 점수화 + 문장 내 공출현 부스트 +0.15 → 합 ≥ 임계값이면 검출
- **Pass B** — 누적 사전으로 약한 후보 재평가
- **동적 임계값** — 기본 0.50, 2자 + 이름끝 약하면 0.60 (엄격)

→ 각 검출은 evidence 필드에 **발동한 신호 목록**을 남겨 추적·감사 가능