

# ⚡ 쇼트컷 (Short-Cut) v3.0

아이디어의 입력만으로 **선행특허 조사**와 **침해 리스크**를 평가하는 AI 솔루션

## Short-Cut 아이디어의 시작

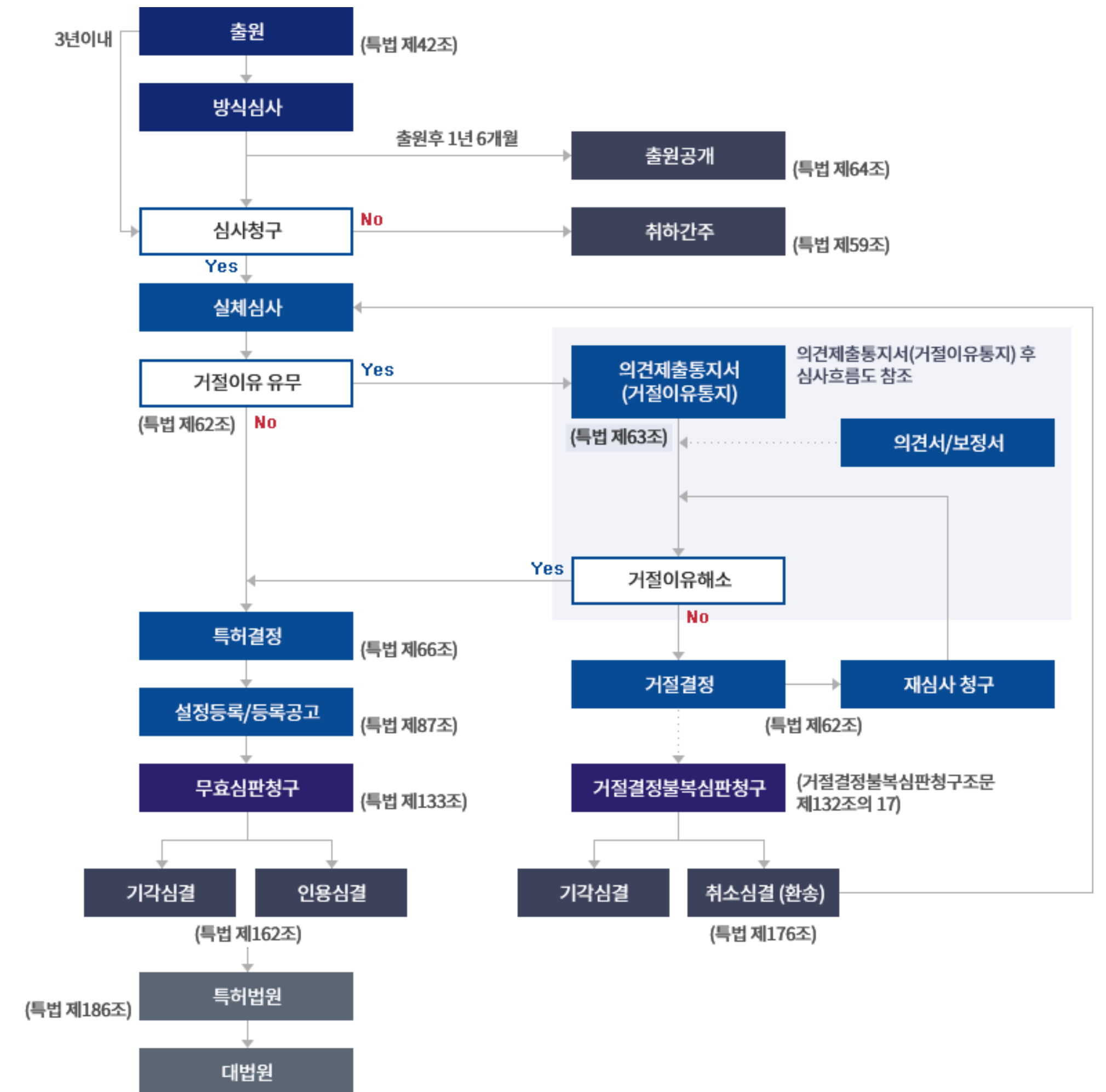
2025년 기준 월 평균  
특허 출원 등록 건수  
약 13,000~15,000건  
(2025년 기준 특허 출원 건수 약 26만 건)



# Short-Cut 아이디어의 시작

## 특허 출원의 복잡한 업무 프로세스

(특허로 [www.patent.go.kr](http://www.patent.go.kr))



## Short-Cut 아이디어의 시작

**변리사를 통해 특허 등록 시  
부가 되는 과도한 비용  
(특허 출원까지 약 500만원)**





**복잡하고 어렵고 돈도 많이 드는데  
해결할 방법은 없는 걸까?**



**쉽고**

혁신을 앞당기는 가장 빠른 지름길

**빠르고**

문턱을 낮춘 가장 쉬운 가이드

**低비용**

가치를 실현하는 합리적인 금액



# 목차

## 01 시스템 개요

- 쇼특허 v3.0 소개
- Self-RAG 기반 특허 분석 시스템 개요

## 02 목표와 가치

- 빠른 의사결정 지원
- 침해 리스크 평가(Claim 관점)
- 회피/차별화 전략 제안

## 03 사용자 플로우

- 입력 → 검색 → 분석 → 출력

## 04 데이터 수집

- 데이터 소스
- 수집 범위
- 대상 국가/기술분야(IPC)

## 05 전처리 파이프라인

- 청구항 파싱(4-Level)
- 임베딩 생성
- 인덱싱 방식(Pinecone + BM25)

## 06 핵심 기술 구성

- 하이브리드 검색(Hybrid Search)
- 리랭커(Reranker) 정밀 재정렬
- 청구항 단위 분석(Claim-Level)

## 07 정리 및 다음 단계로

- 시스템 아키텍처 및 모듈 구성
- 한계 및 향후 개선 방향

# 시스템 개요

## 우리가 해결하려는 것

- ✓ 자연어 아이디어 입력만으로 유사 특허 후보(Top-K)를 빠르게 검색
- ✓ 청구항/구성요소 단위로 침해 위험을 자동 점검
- ✓ 선정 근거(인용 구간/랭킹/필터)와 함께 결과를 스트리밍으로 즉시 제공

## 쇼특허(Short-Cut) v3.0 정의

- ✓ 사용자 아이디어를 기준으로 선행특허를 검색·비교해 리스크를 빠르게 점검하는 AI 기반 시스템
- ✓ Self-RAG 기반 검색 고도화(멀티쿼리·HyDE·쿼리 재작성) + 하이브리드 검색(Dense+Sparse) 결합
- ✓ IPC 필터링 → 통합 랭킹(RRF) → Reranker → LLM 요약/분석까지 연결한 엔드투엔드 프로토타입

## 핵심 산출물

- ✓ 유사 특허 후보(Top-K) 및 선정 근거 요약
- ✓ 침해 리스크 평가: 위험 청구항/구성요소 매칭 포인트 정리
- ✓ 구성요소 기반 회피·차별화 전략 + PDF 리포트/시각화(Guardian Map 등) 제공

※ 청구항: 특허가 법적으로 보호받는 권리 범위를 정의한 문장



# 목표와 가치

쇼특허(Short-Cut) v3.0는 출원/제품화 단계에서 사용자의 아이디어가 기존 특허와 얼마나 유사한지, 침해 리스크(청구항/구성요소)가 어디에 있는지, 그리고 어떤 방식으로 회피·차별화할지까지 빠르게 판단하도록 돕는 AI 기반 선행기술 분석 시스템입니다.

## 신속한 유사 후보 선정

- 사용자 아이디어 입력만으로 유사 특허 후보(Top-K) 신속 제시
- 하이브리드 검색(Dense+Sparse) + IPC 필터링으로 관련 후보 압축
- Reranker로 핵심 후보 우선 도출

## 침해 리스크 평가 (Claim 관점)

- 청구항(Claim)·구성요소 단위로 매칭하여 침해 가능성 점검
- All Elements Rule 관점으로 충족/누락 요소를 명확히 구분
- 위험 청구항/구성요소를 근거(인용 구간/요약)와 함께 제시

## 회피/차별화 전략 제안

- 아이디어 vs 특허 구성요소 대비표 제공
- 충돌(위험) 요소 중심으로 설계 변경·대체안 제안
- 차별화 포인트 및 개발 방향을 리포트(PDF/시각화) 형태로 정리

※ 청구항: 특허가 법적으로 보호받는 권리 범위를 정의한 문장

# 사용자 플로우

## 입력 단계

- 사용자가 아이디어(기술 설명)를 자연어로 입력
- 필요 시 관심 기술 분야 IPC 필터 선택
- 검색 범위/관점(키워드, 목적, 비교 대상 특허 등) 설정

## 검색 단계

- 아이디어를 다양한 관점으로 확장(멀티쿼리 생성, HyDE/쿼리 재작성)
- 하이브리드 검색: Dense(의미 기반) + Sparse(BM25 키워드)

## 분석 단계

- Reranker(Cross-Encoder)로 상위 후보 정밀 재정렬
- 상위 특허에 대해 청구항(Claim)·구성요소 단위로 비교/매칭
- 위험 포인트(침해 가능성)와 근거(인용 구간/요약) 정리

## 출력 단계

- 결과를 스트리밍 방식으로 실시간 제공
- 유사 특허 후보 리스트(Top-K) + 선정 근거
- 침해 리스크 리포트 + 회피/차별화 가이드(PDF/시각화 포함)

# 데이터 수집

데이터 소스	Google Patents Public Dataset (BigQuery)
수집 기간	2018-01-01 ~ 2024-12-31
대상 국가	US, EP, WO, CN, JP, KR
수집량	10,000건(데모/프로토타입 목적)
도메인/IPC	AI/NLP 도메인 키워드 기반 선별 + IPC: G06F 16, G06F 40, G06N 3/5/20, H04L 12

# 전체 파이프라인

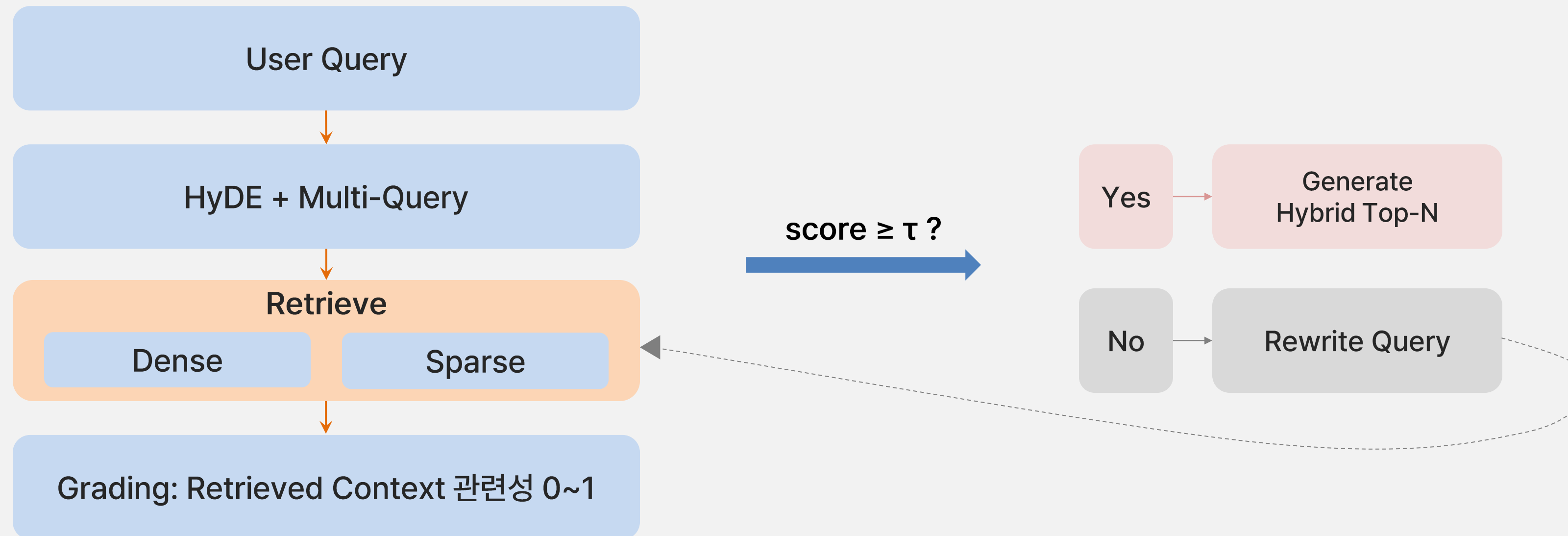
1	데이터 추출	<ul style="list-style-type: none"><li>BigQuery에서 특허 데이터 추출</li></ul>
2	전처리	<ul style="list-style-type: none"><li>청구항 파싱 (독립항/종속항)</li><li>텍스트 청킹 (512 tokens), RAG 컴포넌트 태깅</li></ul>
3	Triplet 생성	<ul style="list-style-type: none"><li>PAI-NET 학습용 triplet 생성 (Anchor: 피인용 특허, Positive: 인용특허, Negative: 무관한 특허)</li></ul>
4	임베딩	<ul style="list-style-type: none"><li>OpenAI API로 벡터 임베딩 생성</li></ul>
5	인덱싱	<ul style="list-style-type: none"><li>Pinecone에 벡터 업로드</li></ul>
6	Self-RAG 데이터	<ul style="list-style-type: none"><li>Ground Truth 데이터 생성</li></ul>

특허 원천데이터를 정제하고(청킹/태깅), 학습·평가 데이터(triplet/GT)를 구축한 뒤 임베딩·인덱싱까지 연결해 검색 기반을 완성했습니다.



# 핵심 기술 구성 (1)

## Hybrid Search



### Self-RAG + Grading/Rewrite Loop

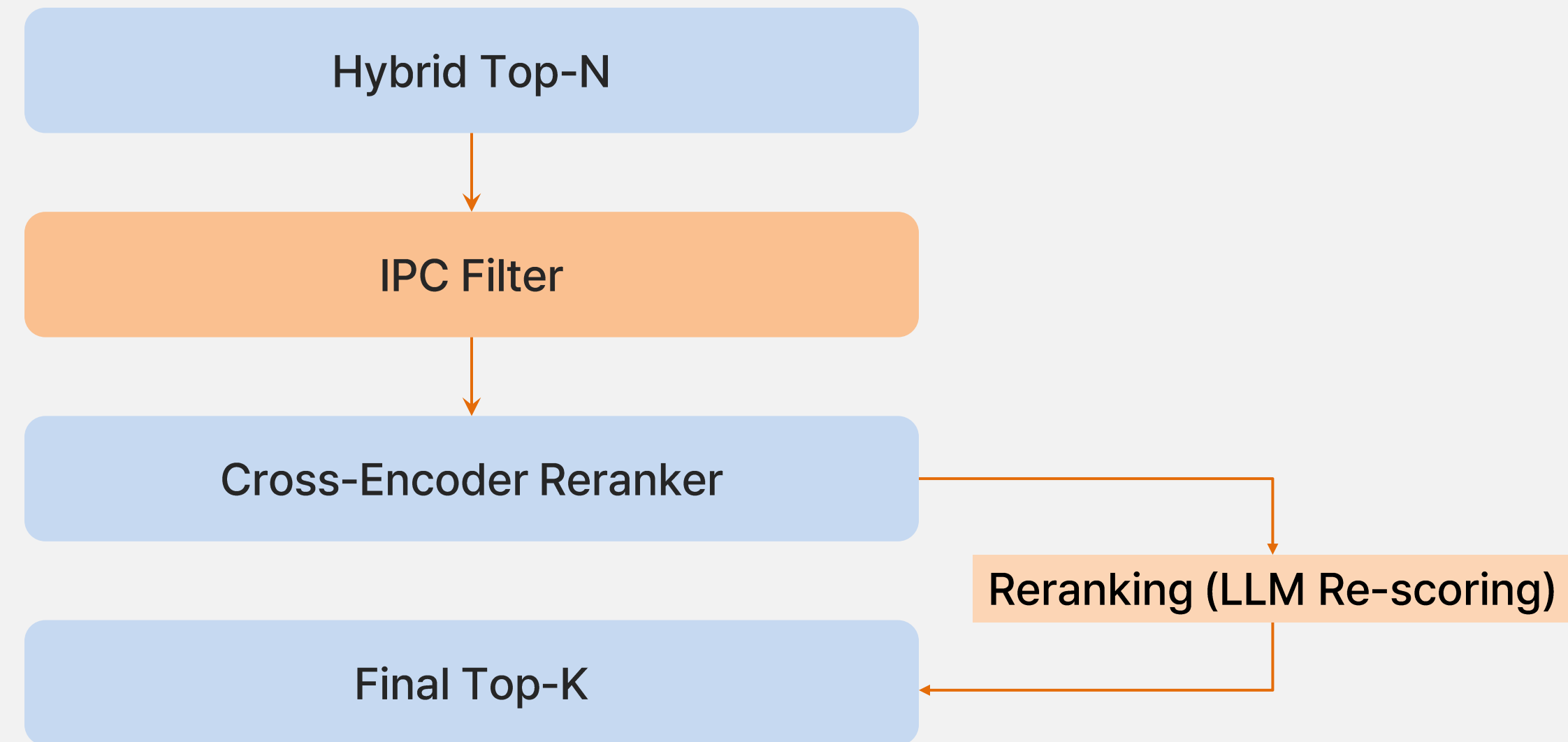
- 검색으로 가져온 후보 문서/청구항(Context)에 관련성 점수(0~1) 부여
- 점수가 임계값( $\tau$ ) 미만이면 질의 재작성 후 재검색(최대 1회)
- 통과 시 다음 단계(분석/요약)로 진행해 저품질 검색 결과를 자동 차단
- Dense: Pinecone 벡터 검색, Sparse: BM25 키워드 검색

### HyDE & Multi-Query RAG

- HyDE: 입력 아이디어로 '가상 특허 설명/청구항 형태' 생성 → 검색어 강화
- Multi-Query: 관점별 질의 N개 확장 → Recall/커버리지 개선
- 확장된 질의 결과를 통합해 후보 품질 향상

## 핵심 기술 구성 (2)

### IPC Filtering & Reranker



- 사용자가 선택한 IPC(기술 분류)를 기준으로 검색 결과를 1차 필터링
- Cross-Encoder 기반 Reranker로 상위 후보를 정밀 재정렬
- Reranking: 1차 검색된 상위 후보군을 고성능 모델(gpt-4o)로 재채점
- 최종적으로 분석 가치가 높은 Top-K (K=5)를 선별

## 핵심 기술 구성 (3)

### Claim-Level Analysis

- All Elements Rule 기반으로 청구항(Claim) 침해 가능성 평가
- 문서 유사도(abstract) 중심이 아닌, 구성요소 단위로 세부 매칭/비교
- 위험 청구항 및 침해 가능 포인트 도출(충족/누락 요소 구분)
- 아이디어 ↔ 청구항 구성요소 매칭/차이점 정리
- 구성요소 대비표 자동 생성 + 회피/차별화 방향 제안

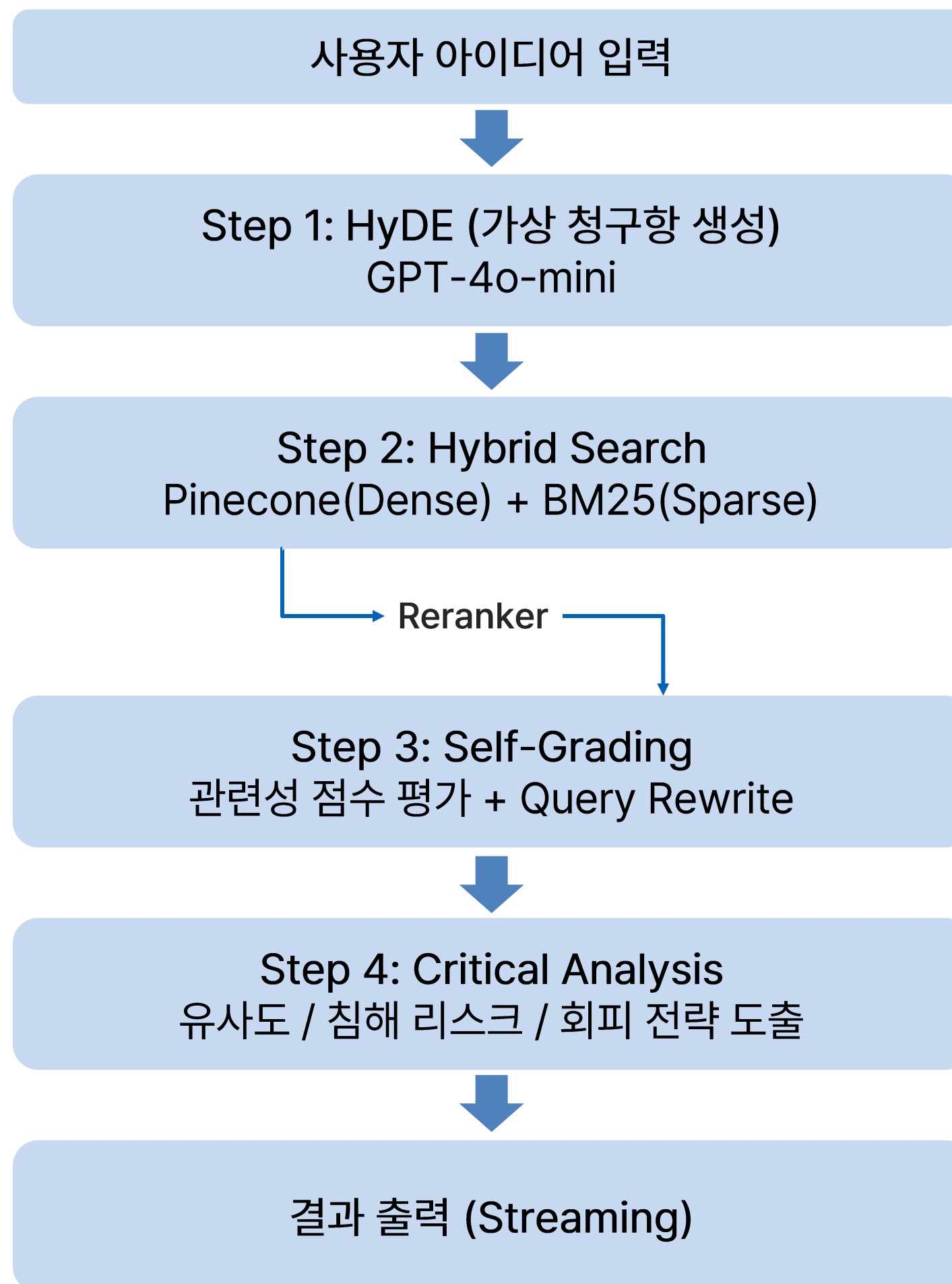


### 피드백 & 시각화

- 사용자 피드백을 로그로 수집하여 품질 개선에 활용
- 피드백 데이터/분석 결과를 이력(SQLite)으로 관리(재조회·비교 가능)
- LLM 결과를 스트리밍 방식으로 즉시 출력(대기시간 감소)
- Guardian Map(성/침입자) + 특허 지형도 시각화로 후보군을 직관적으로 제공
- PDF 리포트 자동 생성(Top-K, 위험 청구항/구성요소, 회피·차별화 전략 포함)

# 정리 및 다음 단계로

## 시스템 아키텍처 및 모듈 구성



### Streamlit(app.py)

- 아이디어 입력/IPC 선택/실행 버튼 등 사용자 입력 처리
- 검색·분석 결과 스트리밍 출력 및 화면 구성
- 후보 리스트/리포트/시각화 화면 렌더링

UI 레이어

### patent\_agent.py, vector\_db.py, reranker.py

- Multi-Query/HyDE 기반 쿼리 생성 및 오케스트레이션
- Hybrid 검색(Dense + Sparse/BM25) + IPC 필터링
- Cross-Encoder 기반 정밀 재정렬(Rerank) 및 Top-K 선별

핵심모듈

### analysis\_logic.py, session\_manager.py, preprocessor.py, embedder.py, feedback\_logger.py, history\_manager.py, ui/\*

- 청구항 파싱/청킹 등 전처리 파이프라인 및 임베딩 생성
- 인덱싱/검색 설정 지원 및 결과 후처리(근거 정리/요약 입력 구성)
- 사용자 피드백 로깅 및 분석 이력(SQLite) 관리(재조회·비교)
- UI 공통 컴포넌트 분리 및 상태(Session) 관리

데이터 처리 & 사용자 경험



# 정리 및 다음 단계로

## 한계 및 향후 개선 방향

현재 시스템은 데모용 10K 특허 데이터 기반으로 구축되어, 전체 특허 커버리지 및 도메인 확장에 한계가 있습니다.

- **QA 현황:** pytest 32개 테스트 100% Pass / Golden(DeepEval) Pass Rate 83.1%(69/83)
- **데이터 확장:** 전체 특허 DB/도메인 확장으로 검색 커버리지 개선
- **비용/안정성:** OpenAI API Mock + 캐싱으로 비용 절감 및 재현성 강화
- **검색 고도화:** IPC 필터 정교화 + Hybrid(Dense+BM25) 파라미터 튜닝
- **피드백 기반 고도화:** 사용자 피드백 로그를 활용한 Reranker/검색 전략 개선
- **산출물 고도화:** PDF 리포트/시각화(Guardian Map 등) 품질 및 자동화 범위 확대