□ 쇼특허(Short-Cut) v3.0

아이디어의 입력만으로 선행특허 조사와 침해 리스크를 평가하는 AI 솔루션

Team: 뀨□

목차

시스템 개요

우리가 해결하려는 것

- ✓ 자연어 아이디어 입력만으로 유사 특허 후보(Top-K)를 빠르게 검색
- ✓ 유사 특허를 근거로 침해 리스크(청구항/구성요소 관점)와 위험 포인트를 요약
- ✓ 결과는 스트리밍으로 빠르게!

쇼특허(Short-Cut) v3.0 정의

- ✓ 사용자 아이디어를 기준으로 선행특허를 검색·비교해 리스크를 빠르게 점검하는 AI 기반 시스템
- ✓ Self-RAG 기반 검색 고도화 + LLM 요약을 결합한 선행기술 조사 프로토타입

핵심 산출물

- ✓ 유사 특허 후보(Top-K) 및 선정 근거 요약
- ✓ 침해 리스크 평가: 위험 청구항/구성요소 매칭 포인트 정리
- ✓ 구성요소 대비표 + 회피·차별화 전략 제안

목표와 가치

쇼특허(Short-Cut) v3.0은 출원/제품화 전 단계에서 사용자의 아이디어가 기존 특허와 얼마나 유사한지, 침해 리스크가 어디에 있는지, 그리고 어떻게 회피·차별화할지를 빠르게 판단할 수 있도록 돕는 AI 기반 선행특허 분석 시스템입니다.

빠른 의사결정 지원	- 사용자 아이디어와 유사한 특허 후보(Top-K) 제시 - 의미 기반 검색(Dense)와 키워드 검색(Sparse)을 결합 - 정밀 재정렬(Rerank)로 핵심 후보를 우선 도출
침해 리스크 평가 (Claim 관점)	- 특허 청구항(Claim) 기준으로 침해 가능성 점검 - '모든 구성요소 법칙(All Elements Rule)' 관점으로 평가 - 위험 청구항/구성요소를 근거와 함께 명확히 제시
회피/차별화 전략 제안	- 청구항 구성요소 대비표 제공(아이디어 vs 특허) - 충돌 지점(위험 포인트) 중심으로 개선 방향 정리 - 설계 변경/차별화 방향을 요약 제안

※ 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 등) 설정

검색 단계

- 아이디어를 다양한 관점으로 확장(멀티 쿼리 생성)
- 하이브리드 검색: Dense(의미 기반) + Sparse(BM25 키워드)
- 결과 융합(RRF)으로 후보군 통합 및 중복 제거

분석 단계

- Reranker(Cross-Encoder)로 상위 후보 정밀 재정렬
- 상위 특허에 대해 청구항(Claim) 단위로 구성요소 비교
- 위험 포인트(침해 가능성) 및 근거 요약

출력 단계

- 결과를 스트리밍 방식으로 실시간 제공
- 유사 특허 후보 리스트(Top-K)
- 침해 리스크 평가 리포트 + 회피/차별화 가이드

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 관련, G06F16, G06F40, G06N 계열

데이터 전처리 파이프라인

데이터 추출

- BigQuery에서 특허 데이터 추출 (2018–2024)
- 대상: US / EP / WO / CN / JP / KR
- 텍스트 정규화(특수문자/공백/인코딩 처리)

청구항 파싱

- 4-Level 청구항 파싱 적용(구성요소 단위 분해)
- 국가/형식 차이를 고려한 파싱 규칙 적용(US/EP/KR 등)
- 괄호/대괄호 번호 등 표현 정리(전처리 규칙)

청킹 & 임베딩

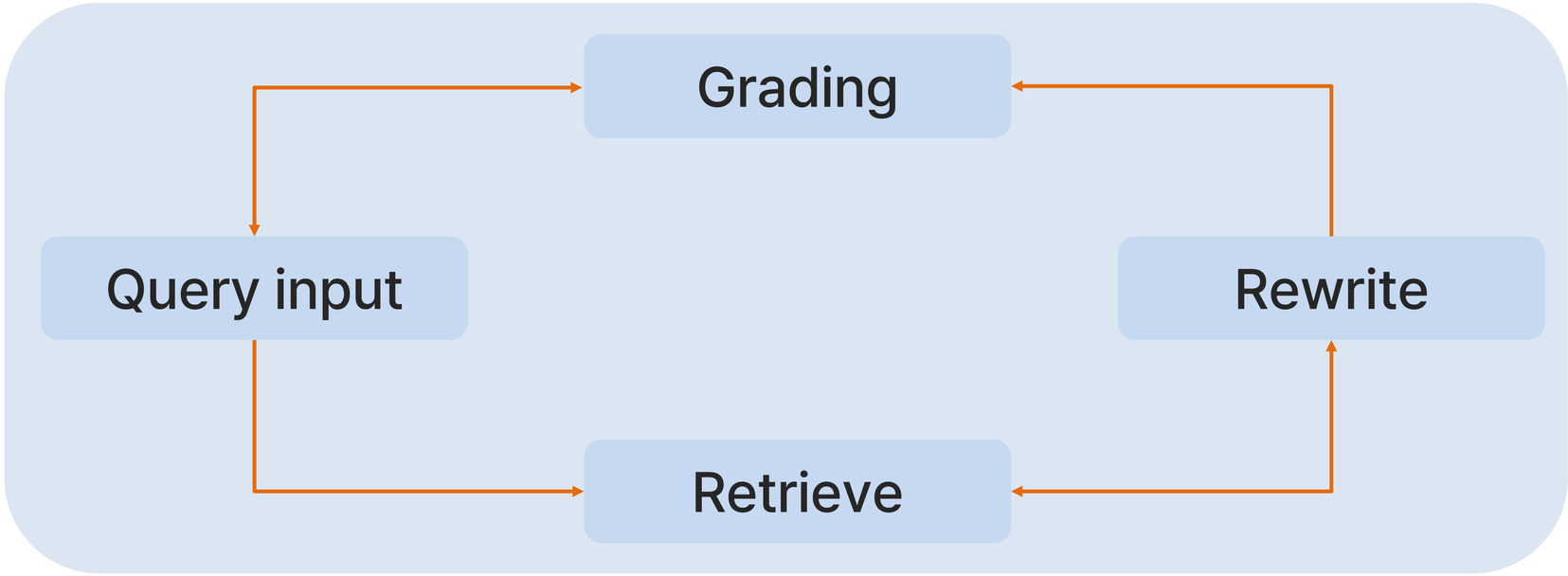
- 최대 1024 토큰 단위로 청킹
- 오버랩 128 토큰 적용(문맥 유지)
- 임베딩: text-embedding-3-small (1536차원)

인덱싱

- Dense: Pinecone(Vector DB) 서버리스 인덱스
- Sparse: 로컬 BM25 인덱스
- 하이브리드 검색 기반 구성(Density + Keyword)
- 약 20,664 청크/벡터 생성

데이터 전처리는 검색 품질을 좌우하는 핵심 단계입니다.
BigQuery 데이터는 청구항 4-Level 파싱 → 청킹/임베딩을 거쳐 의미 기반 검색이 가능해지고,
Pinecone(Dense) + BM25(Sparse) 이중 인덱싱으로 하이브리드 검색 기반을 마련합니다.

핵심 기술 구성 (1)



Self-RAG + Grading/Rewrite Loop

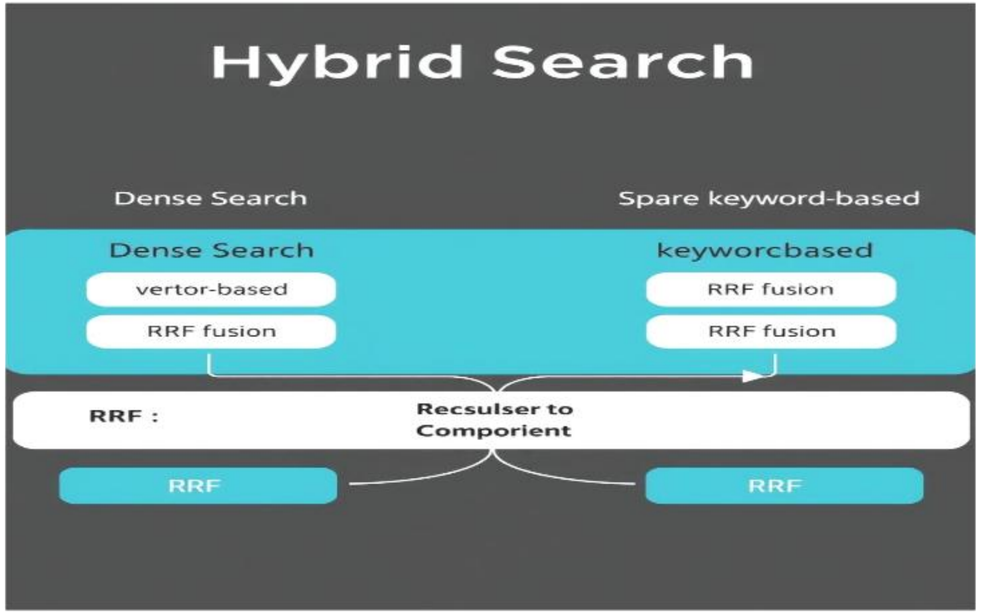
- 검색 결과에 대해 0~1 관련성 점수(Grading)를 부여
- 평균 점수가 임계값(예: 0.6) 미만이면 쿼리 재작성(Rewrite) → 재검색(Retrieve) 1회 수행
- 저품질 검색 결과를 줄여 분석 품질을 안정화하는 루프

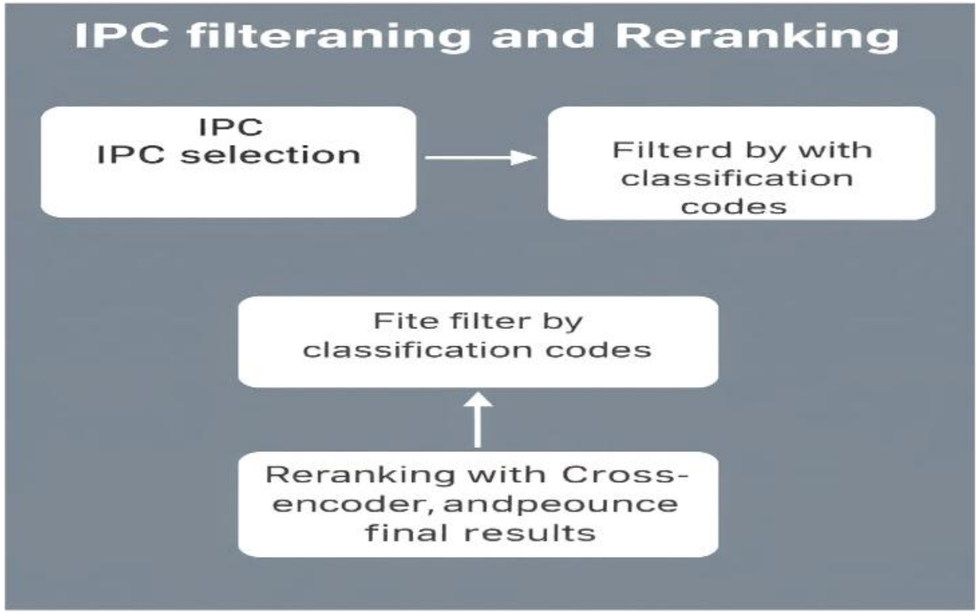
HyDE & Multi-Query RAG

- HyDE: 아이디어로부터 가상의 특허 설명/청구항 형태의 문장을 생성해 검색 쿼리를 강화
- Multi-Query: 기술/청구항/문제해결 관점 등 여러 쿼리로 확장해 검색 커버리지 개선
- 짧거나 애매한 입력에서도 검색 품질·재현율(Recall)을 높이는 목적

핵심 기술 구성 (2)

GPT가 그려준 모식도인데, 맞는지 잘 모르겠습니다.





Hybrid Search + RRF

- Dense(의미 기반) + Sparse(키워드 기반) 검색을 결합해 후보를 넓게 확보
- Dense: Pinecone 벡터 검색, Sparse: BM25 키워드 검색
- RRF(Reciprocal Rank Fusion)로 두 결과를 통합해 상위 후보를 안정적으로 선정
- 파라미터: k=60, dense/sparse 가중치 0.5 : 0.5

IPC Filtering & Reranker

- 사용자가 선택한 IPC(기술 분류)를 기준으로 검색 결과를 1차 필터링
- Cross-Encoder 기반 Reranker로 상위 후보를 정밀 재정렬
- 모델: ms-marco 계열 활용
- 최종적으로 분석 가치가 높은 Top-K를 선별

핵심 기술 구성 (3)

Claim-Level Analysis

- 모든 구성요소 법칙(All Elements Rule) 기반 청구항(Claim) 분석
- 문서 유사도 대신 구성요소 단위로 세부 비교
- 위험 청구항 및 침해 가능 포인트 도출
- 아이디어 ↔ 청구항 구성요소 매칭/차이점 정리
- 구성요소 대비표 자동 생성 + 회피/차별화 방향 제안

피드백 & 시각화

- 사용자 피드백을 로깅하여 개선에 활용
- 피드백 데이터를 Reranker 학습 데이터로 축적(향후 고도화)
- SQLite 기반 분석 히스토리 관리(재조회/비교 가능)
- LLM 결과를 스트리밍 방식으로 즉시 출력(사용자 대기 감소)
- 특허 지형도 시각화(거리/연결선 등)로 유사 특허 군집을 직관적으로 제공

정리 및 다음 단계로

시스템 아키텍쳐 및 모듈 구성

Streamlit(app.py) – UI 레이어

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

patent_agent.py, vector_db.py, reranker.py – 핵심 모듈

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

preprocessor.py, embedder.py, feedback_logger.py, history_manager.py, ui/*

- 데이터 처리 & 사용자 경험

- 청구항 파싱·청킹 등 전처리 파이프라인
- 임베딩 생성 및 인덱싱 지원
- 사용자 피드백 로깅 및 분석 이력 SQLite 관리
- UI 컴포넌트/공통 화면 요소 분리

정리 및 다음 단계로

한계 및 향후 개선 방향

현재 시스템은 데모용 10K 특허 데이터 기반으로 구축되어, 전체 특허 범위를 포괄하는 데 한계가 있습니다.

- 데이터 확장: 대규모 특허 DB 구축(커버리지 개선)
- 비용/안정성 개선: OpenAI API Mock 도입 및 캐싱 적용
- 로컬 인덱스 고도화: FAISS 등 로컬 인덱스 연동·I/O 테스트 강화
- 테스트 자동화: Streamlit E2E(UI) 테스트 자동화
- 피드백 기반 고도화: Reranker/검색 전략을 사용자 피드백으로 개선