뤼튼(wrtn) AX팀

Agent Developer
인턴 과제 발표

sLLM 기반 RAG Agent 성능 향상 프로젝트

KMMLU Criminal-Law & Law 도메인 최적화를 위한
엔지니어링 접근과 10가지 실험의 기록

최종 성과: Baseline 51.58% → 56.76% (+5.18%p)

발표자

지원자

2026.01.28

Agent Engineering Assessment

과제 해석 및 접근 전략

과제를 받고 든 생각

STARTING POINT

"왜 이런 과제를 주셨을까?"

저의 가설

sLLM과 Agent 구조로
효율적인 고성능을 달성하는
엔지니어링 사고력 평가

과제 요구사항에서 발견한 단서들

모델 리소스 제한

gpt-4o-mini, embedding-small만 허용

→ 제한된 리소스로 최선을 찾아라

Production 수준 시스템

단순 1회성 정답 맞추기가 아님

→ 단순 정확도보다 견고한 시스템 설계 중요

Baseline 점수 제공

Dev set 기준 51.58%

→ 이 점수를 넘어서는 것이 1차 목표

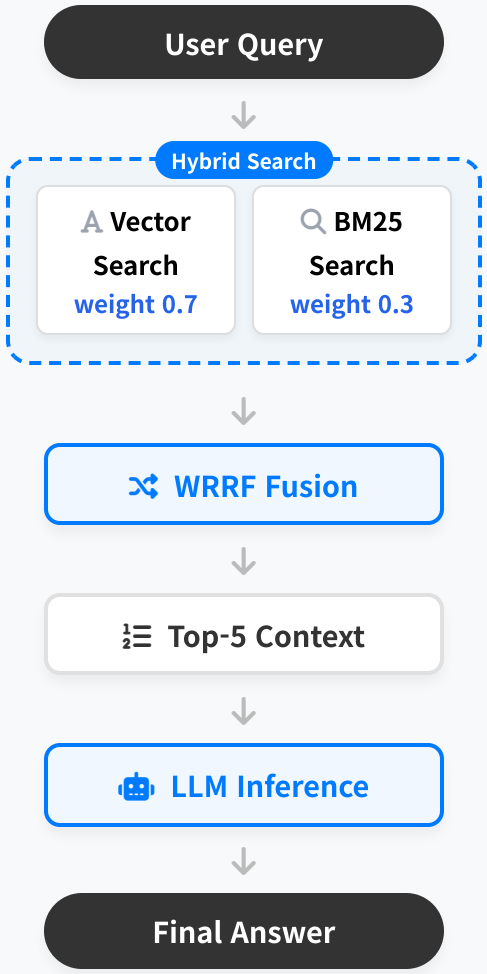
Action Plan

가설 검증을 위해 10개의 실험 수행

SYSTEM ARCHITECTURE

Agent 시스템 구조

효율적인 검색과 생성을 위한 RAG 파이프라인 및 핵심 기술 스택



핵심 컴포넌트 상세 (Core Components)

KNOWLEDGE BASE
train.csv 기반 ChromaDB 2,073 chunks

RETRIEVAL STRATEGY
Hybrid Search (Vector 0.7 + BM25 0.3)

FUSION ALGORITHM
WRRF (Weighted Reciprocal Rank Fusion)

TOKENIZER
kiwipiepy (한국어 형태소 분석기)

LLM ENGINE
gpt-4o-mini seed=42

ORCHESTRATION
LangGraph Framework

DESIGN DECISION

초기 설계 의사결정

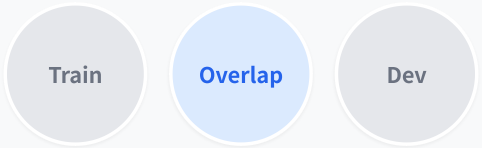
데이터 분석 및 질문

"왜 train.csv를 Knowledge Base로?"

기존 KMMLU 데이터와 다른 점을 발견하여, LLM을 활용해 train 데이터와 dev 데이터를 교차 분석했습니다.

분석 결과 발견점

train.csv와 dev.csv에 **겹치는 문제(Duplicate/Overlap)**가 일부 존재함을 확인했습니다.



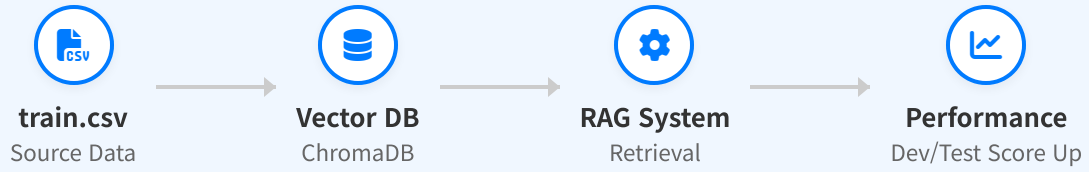
가설 수립 및 근거

HYPOTHESIS

"train.csv를 RAG Knowledge Base로 활용하면, dev.csv 성능 향상을 이끌어낼 수 있을 것이다."

1. **겹치는 문제 존재**
   동일/유사 문제는 Retrieval 시 직접적인 정답 힌트로 작용
2. **동일 출처 패턴 (KMMLU)**
   겹치지 않는 문제라도 문체나 논리 구조가 유사하여 Few-shot 효과 기대
3. **RAG 활용 의도 추정**
   train 데이터를 학습(Fine-tuning)이 아닌 RAG DB로 쓰라는 의도로 해석

결과: 가설 적중 (성능 향상 확인)



EXPERIMENT OVERVIEW

실험 전체 조감도

Baseline(51.58%) 극복을 위한 10번의 실험과 성능 추이

실험 요약

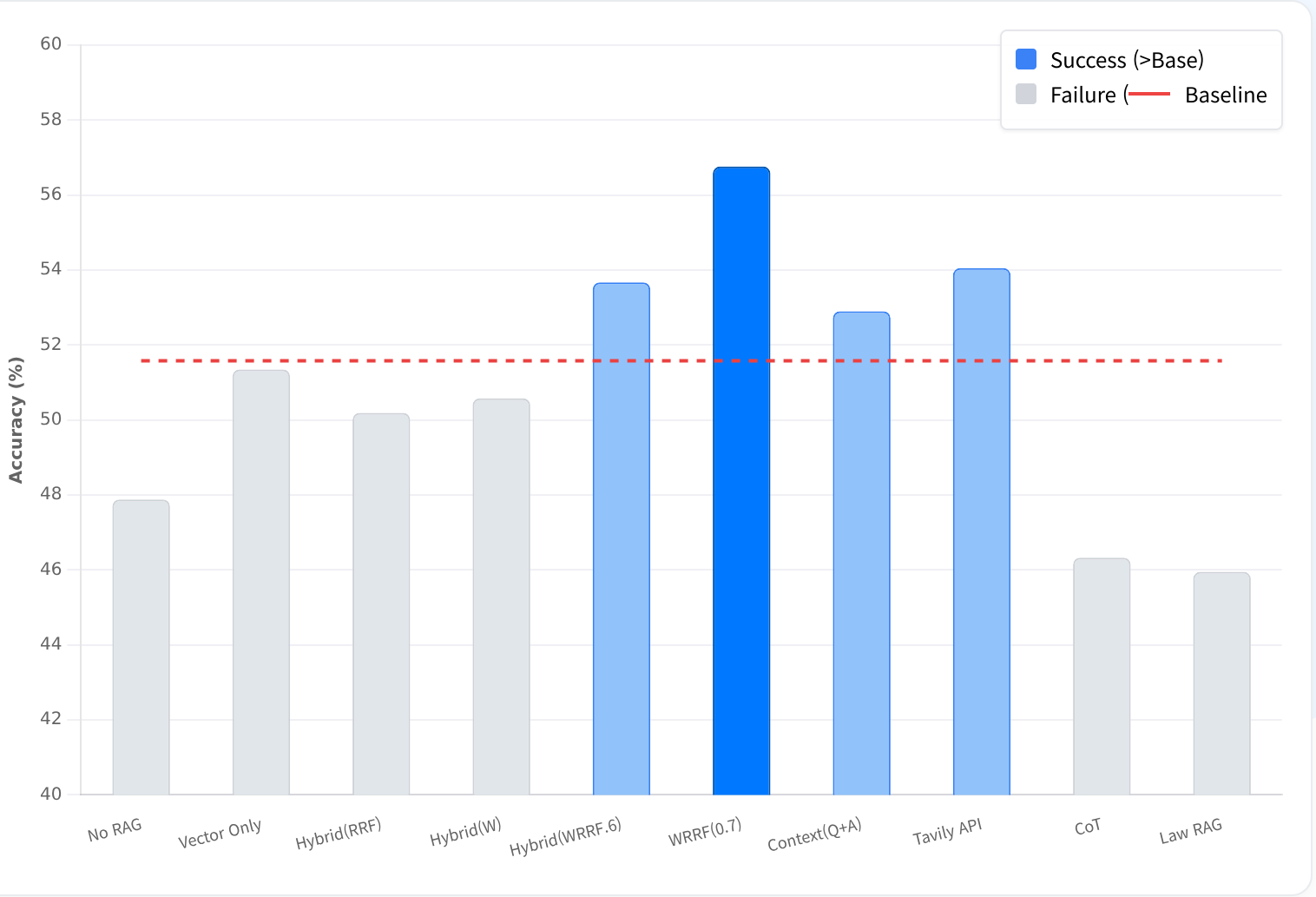
총 실험 횟수	10회
Baseline (Dev)	51.58%
최고 성능 (Best)	56.76%
성능 향상폭	+5.18%p

MAJOR FINDINGS

- **Hybrid Search + WRRF**
  가장 안정적이고 높은 성능 기록
- **CoT & 외부 검색**
  복잡도 증가 대비 성능 하락

KEY INSIGHT

"복잡한 기법보다
데이터와 검색의 기본기가
성능을 결정했습니다."



SUCCESSFUL EXPERIMENT

성공한 실험 ① Hybrid Search 진화기

검색 전략의 시행착오 끝에 찾아낸 최적의 WRRF 파라미터 튜닝 과정

1. Vector Only	2. RRF (k=60)	3. Weighted	4. WRRF (0.6)	5. WRRF (0.7)
51.35%	50.19%	50.58%	53.67%	56.76%
Baseline	▼ 1.16%p	▼ 0.77%p	▲ 2.32%p	▲ 5.41%p

실험 과정에서의 배운 점

- Vector 검색만으로는 법률 전문 용어 매칭에 한계가 있음 (단순 RRF는 오히려 성능 저하)
- WRRF 는 순위(Rank)의 안정성과 가중치(Score)의 중요도를 동시에 고려하여 가장 효과적

최종 튜닝 성과

Vector 0.7 : BM25 0.3 비율 최적화

0.6:0.4 비율 대비 **+3.09%p** 추가 성능 향상을 이끌어내며,
최종적으로 Baseline 대비 **+5.18%p** 달성

FAILED EXPERIMENT ANALYSIS

실패한 실험 ① Context 형식 실험

"오답 선택지를 제거하면 검색 성능이 올라가지 않을까?"라는 가설의 검증

HYPOTHESIS

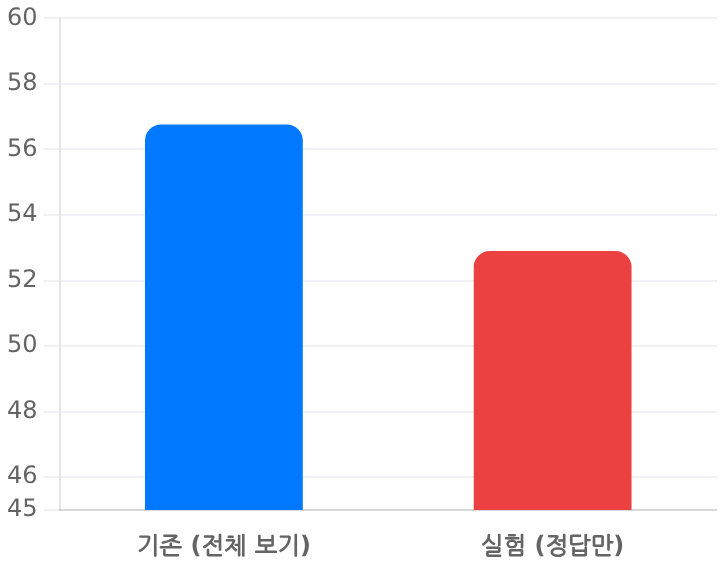
오답 선택지는 노이즈(Noise)가 아닐까?

"Vector Search 시 오답 텍스트가 의미적 유사도 계산을 왜곡하고, BM25 매칭에 불필요한 키워드를 제공할 것이다."

EXPERIMENT SETUP

Q + 정답 보기만 남기고
오답 3개는 제거하여 DB 구축

RESULT DATA



↓ 3.86%p 하락

오답을 제거했더니 오히려 성능이 떨어짐

WHY FAILED?

× LLM 추론 정보 부족

LLM이 정답을 판단할 때, 오답 선택지와의 **비교 정보**가 중요한 힌트로 작용했던 것으로 보임.

× BM25 성능 저하

정답만 남겼을 때 Vector 가중치를 높이면 일부 효과가 있었으나, **BM25 키워드 매칭** 성능이 급격히 떨어짐을 확인.

LESSON LEARNED

기존 형식 (Q + 4선택지 + 정답) 유지 결정

FAILED EXPERIMENT ANALYSIS

실패한 실험 ② 외부 웹 검색

"최신 법령과 판례를 웹에서 검색하면 성능이 개선될까?" (Tavily API)

HYPOTHESIS

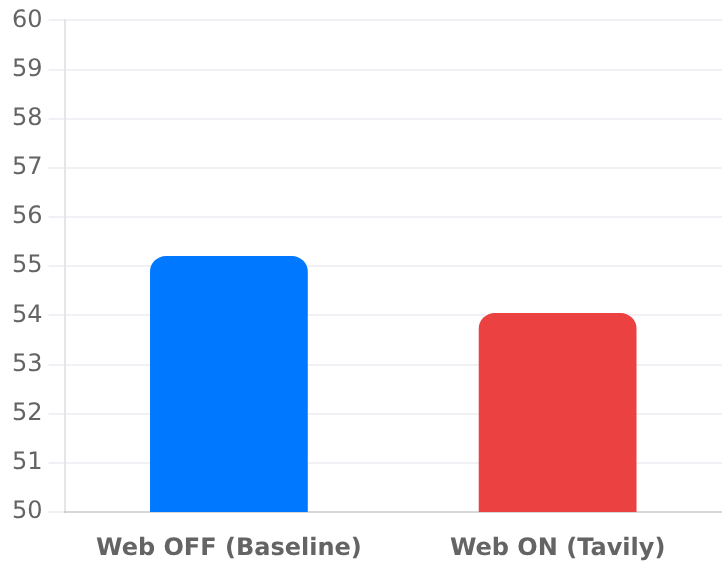
웹 검색이
부족한 지식을 채워줄까?

"정적인 Knowledge Base의 한계를 넘어, 동적인 웹 검색이 최신 법령 정보를 보완해 줄 것이다."

EXPERIMENT SETUP

Tavily API 통합
Context에 검색 결과 추가

RESULT DATA



↓ 1.16%p 하락

웹 검색을 활성화했더니 성능이 오히려 떨어짐

WHY FAILED?

× 데이터 품질 저하

법률은 매우 전문적인 도메인입니다. 일반 웹 문서의 품질은 검증된 내부 데이터(train.csv)보다 **신뢰도와 깊이**가 부족했습니다.

× Context 희석 (Dilution)

검색된 웹 정보가 오히려 핵심 Context(유사 문제)의 비중을 낮추고, 추론을 방해하는 **노이즈(Noise)**로 작용했습니다.

LESSON LEARNED

양질의 내부 데이터가 외부 검색보다 우위

FAILED EXPERIMENT ANALYSIS

실패한 실험 ③ Chain-of-Thought

"단계별 추론(CoT)을 유도하면 더 똑똑해지지 않을까?"라는 가설의 검증

HYPOTHESIS

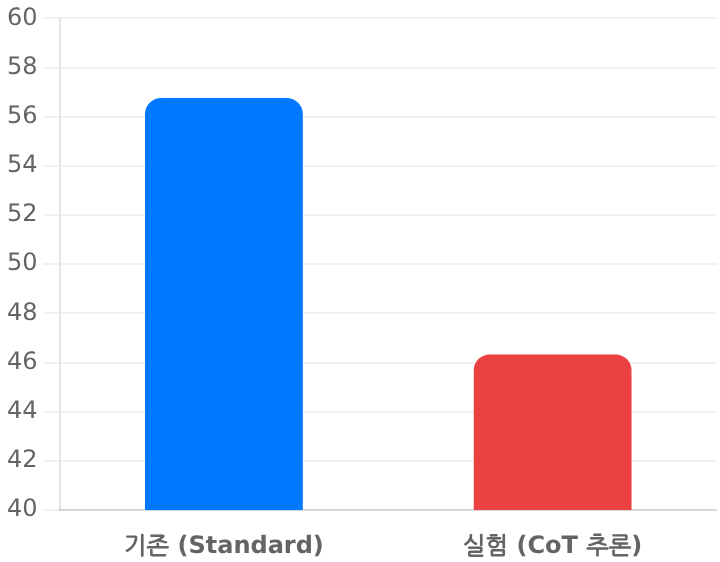
생각의 사슬을 만들면
정답률이 오를까?

"단순히 답을 고르는 것보다, 판례와 법리를 단계적으로 따져보게 하면 복잡한 법률 문제 해결 능력이 향상될 것이다."

EXPERIMENT SETUP

시스템 프롬프트에
"단계별로 생각하고 추론하라"
지시어 추가 (Think step-by-step)

RESULT DATA



↓ 10.43%p 급락

예상과 달리 성능이 크게 떨어짐 (심각한 하락)

WHY FAILED?

× 모델 특성 불일치

gpt-4o-mini 모델은 CoT(Chain-of-Thought) 방식으로 **충분히 학습되지 않았거나**, 경량 모델 특성상 복잡한 추론 과정에서 길을 잃음.

× 도메인 특성

법률 문제는 수학/논리 문제와 달리 계산적 추론보다 **정확한 판례/법령 Context**의 매칭 여부가 정답에 더 결정적임.

LESSON LEARNED

프롬프트 복잡도보다
RAG Context 정확도가 핵심

FAILED EXPERIMENT ANALYSIS

실패한 실험 ④ 법령 문서 RAG

"실제 법령 원문을 검색하면 더 정확해지지 않을까?" → 4차례 시도 끝에 실패

HYPOTHESIS

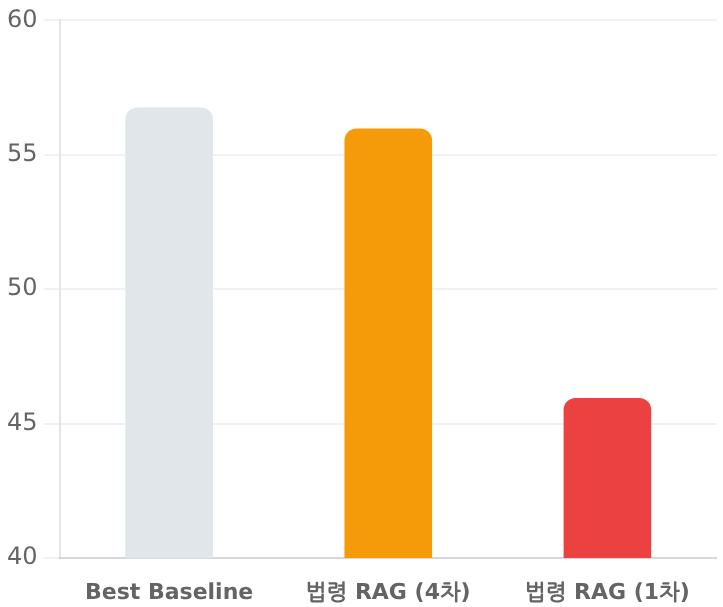
법률 문제의 정답은 법조문에 있다

"기존 Q&A 데이터셋만으로는 한계가 있다. 법제처의 실제 법령(형법 등) 4,000여 개를 지식 베이스에 추가하면 정확도가 오를 것이다."

ATTEMPTS HISTORY

1차	Q&A + 법령 단순 통합 (Top-10)
2차	질문 유형별 DB 라우팅
3차	단일 DB 통합 재구축
4차	조건부 검색 + 이중 DB 구조

RESULT DATA



− 0.78%p 부족

복잡한 이중 DB 구조까지 도입했으나
기존 최고 성능(56.76%)을 넘지 못함 (최종 55.98%)

WHY FAILED?

Context Overload

법령 문서까지 추가하여 Top-K를 늘리자, 소형 모델(gpt-4o-mini)이 **핵심 정보를 놓치는 현상** (Lost in the Middle) 발생.

임베딩 공간 불일치

'질문-답변(Q&A)' 데이터와 '법조문(Statute)' 텍스트의 성격이 너무 달라, **Hybrid Search의 정확도**가 오히려 희석됨.

FUTURE DIRECTION

GraphRAG 도입

또는 법령을 Q&A 형식으로 변환 후 임베딩 시도

KEY LEARNINGS & INSIGHTS

핵심 깨달음

"복잡하다고 좋은 게 아니더라고요" — sLLM 환경에서의 엔지니어링 교훈

sLLM 환경에서 유효했던 것

- **데이터 품질** (Q + 4선택지 + 정답 형식 유지)
- **검색 성능 고도화** (Hybrid Search + WRRF)
- **단순한 아키텍처** (복잡한 라우팅 제거)
- **세밀한 파라미터 튜닝** (0.7 : 0.3)

오히려 해가 됐던 것

- **복잡한 프롬프트** (Chain-of-Thought 강제)
- **외부 지식 검색** (웹 검색 API 무분별한 사용)
- **이질적 데이터 단순 통합** (법령 문서 원문)
- **과도한 Context 확장** (Top-K 무리한 증가)

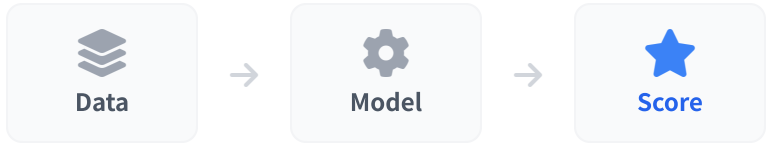
"제한된 리소스 환경에서는
무엇을 **더** 할 것인가보다, 무엇을 **안** 할 것인가가 더 중요했습니다."

FUTURE ROADMAP

추가 개선 방향

시간이 더 있었다면 시도해보고 싶었던
고도화 전략 3가지를 제안합니다.

현재의 Baseline 성능(+5.18%p)에 만족하지 않고, 검색의 정밀도(Precision)와 의미적 일치성(Alignment)을 극대화하기 위한 기술적 접근법들입니다.



Reranker 모델 도입

단기 개선

Hybrid Search로 검색된 Top-50 후보군을 **Cross-Encoder** 모델로 정밀하게 재정렬하여 최종 Top-5의 연관성을 극대화합니다.

Query Expansion (질문 확장)

단기 개선

사용자의 질문을 유사한 의미의 다양한 문장으로 변환(HyDE 등)하여 검색하여, **키워드 불일치 문제**를 완화하고 Recall을 향상시킵니다.

법령 데이터 구조화 (Statute to Q&A)

법령 통합 대안

실패했던 법령 원문 임베딩 대신, LLM을 활용해 법조문을 **질문-답변 쌍**으로 변환한 후 Knowledge Base에 통합합니다.

Before "형법 제250조(살인) ①사람을 살해한 자는..."

↓LLM Transformation

After **"살인죄의 법적 정의와 처벌 규정은 무엇인가?"**

SUMMARY & CONCLUSION

마무리 요약

01. 과제 해석

엔지니어링 사고력 평가

"제한된 sLLM 환경에서 Agent 구조를 활용해
어떻게 **효율적인 고성능**을 달성할 것인가?"

02. 접근 방식

가설 - 실험 - 검증

단순 구현이 아닌, **10번의 체계적인 실험**을 통해
최적의 조합(Hybrid Search + WRRF)을 도출

03. 최종 성과

Baseline 대비 +5.18%p

~~51.58%~~ → **56.76%**

KMMLU Criminal-Law & Law 도메인 정확도 달성

04. 핵심 인사이트

"Back to Basic"

복잡한 프롬프트나 외부 데이터보다
데이터 품질, 검색 정확도, 파라미터 튜닝과 같은
기본기가 성능 향상의 핵심이었습니다.

APPENDIX

예상 Q&A

면접관님의 질문에 대비하여 준비한 예상 질의응답입니다.

Q1 RRF, Weighted, WRRF의 차이가 뭔가요?

WRRF는 RRF의 '순위 기반 안정성'과 Weighted의 '중요도 반영'을 결합한 방식입니다. 점수 스케일이 다른 검색 결과들을 순위로 정규화하면서도, Vector 검색의 중요도(0.7)를 반영하기 위해 선택했습니다.

Q2 왜 CoT(Chain-of-Thought)가 실패했나요?

gpt-4o-mini 모델은 CoT로 충분히 학습되지 않았으며, 법률 도메인은 논리적 계산보다 **정확한 판례 Context 매칭**이 정답에 더 결정적이기 때문입니다. 복잡한 추론보다 정확한 검색이 중요했습니다.

Q3 법령 통합 실패는 어떻게 극복할까요?

딱딱한 법조문을 **'질문-답변(Q&A)' 형식으로 변환**하여 임베딩 공간의 불일치를 해소하거나, **GraphRAG**를 도입하여 법률 조항 간의 인용/참조 관계를 구조적으로 반영해야 합니다.

Q4 WRRF 가중치 0.7:0.3은 어떻게 정했나요?

경험적 튜닝의 결과입니다. 0.6:0.4, 0.8:0.2 등 다양한 비율을 테스트했을 때, Vector(0.7)와 BM25(0.3) 조합이 가장 높은 정확도를 보였습니다. 의미 검색이 키워드 매칭보다 더 중요했습니다.

Q5 실패한 실험들의 가치는 무엇인가요?

"왜 단순한 구조가 좋은지"를 데이터로 증명할 수 있게 되었습니다. 복잡도를 무작정 높이는 것보다, 데이터 품질과 검색 기본기에 집중하는 것이 sLLM RAG의 핵심임을 확인했습니다.

Q6 train.csv를 Knowledge Base로 쓴 이유는?

데이터 분석 결과 train과 dev셋 간에 겹치는 문제가 다수 식별되었습니다. 과제의 의도가 **"주어진 데이터를 최대한 활용하여 성능을 높이는 것"**이라고 판단하여 RAG 지식원으로 채택했습니다.