wrtn.

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

박영기

2026.02.02

Agent Engineering Assessment

과제 해석 및 접근 전략

과제를 받고 든 생각

STARTING POINT

"왜 이런 과제를 주셨을까?"

저의 생각

sLLM기반 Agent 구조의
시스템 성능 개선 능력과
실무 요구사항 대응 역량 평가

과제 요구사항에서 발견한 단서들

모델 리소스 제한

gpt-4o-mini, embedding-small만 허용

→ 제한된 리소스로 최선을 찾아라

실무 수준 요구사항

API DTO 규격, 10분 내 인퍼런스, Docker 서빙

→ 요구사항을 충족하는 시스템 설계 필요

Baseline 점수 제공

Dev set 기준 51.58%

→ 이 점수를 넘어서는 것이 1차 목표

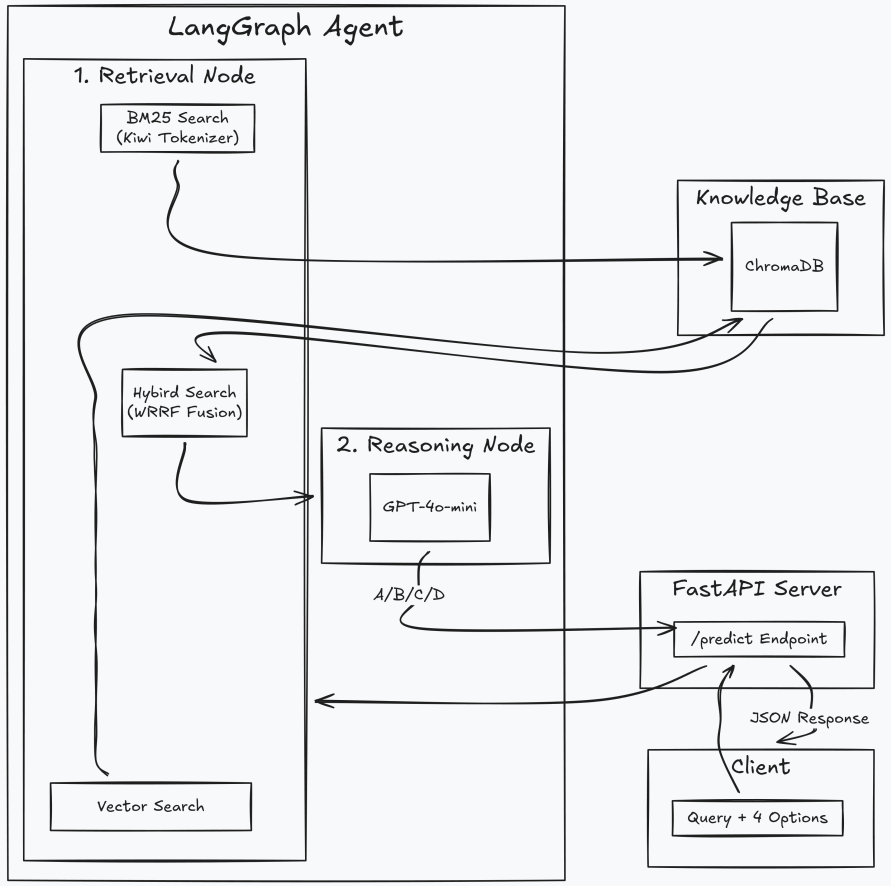
Action Plan

이러한 관점을 중심으로 과제 진행

SYSTEM ARCHITECTURE

Agent 시스템 구조

검색과 생성을 위한 RAG 파이프라인 및 핵심 기술 스택



Retrieval & KB

Hybrid Search + WRRF 방식으로 검색 정확도를 극대화했습니다.
단순 벡터 검색의 한계를 보완하기 위해 키워드 매칭(BM25)을 결합했습니다.

Vector (0.7) | text-embedding-3-small | BM25 (0.3) | Kiwi 형태소 분석 | ChromaDB

Workflow & Serving

LangGraph를 통해 검색과 추론 과정을 노드 단위로 제어합니다. 최종 답변은 JSON 형태로 정규화되어 FastAPI를 통해 서비스됩니다.

LangGraph | LangChain | gpt-4o-mini | FastAPI | Pydantic

DESIGN DECISION

초기 설계 의사결정

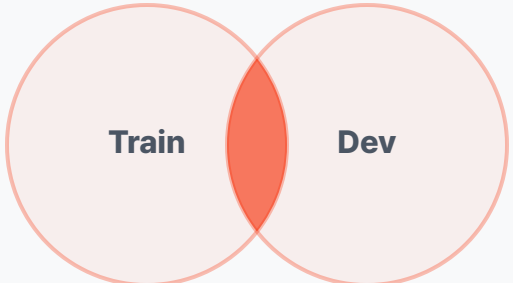
데이터 분석 및 질문

"왜 train.csv를 Knowledge Base로?"

기존 KMMLU와 다른 점을 발견하여,
LLM을 활용해 train.csv와 dev.csv를 분석했습니다.

분석 결과 발견점

train.csv와 dev.csv에 **완전히 동일한 문제가 3개 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. **원본과 다른 문서 활용 의도 추정**
   train.csv를 Vector DB로 사용하라는 의도로 해석

NoRAG (프롬프트만 적용)

47.88%

Vector DB 적용

+3.47%p

RAG 적용(Only Vector)

51.35%

EXPERIMENT OVERVIEW

실험 전체 조감도

실험 과정과 성능 추이

실험 요약

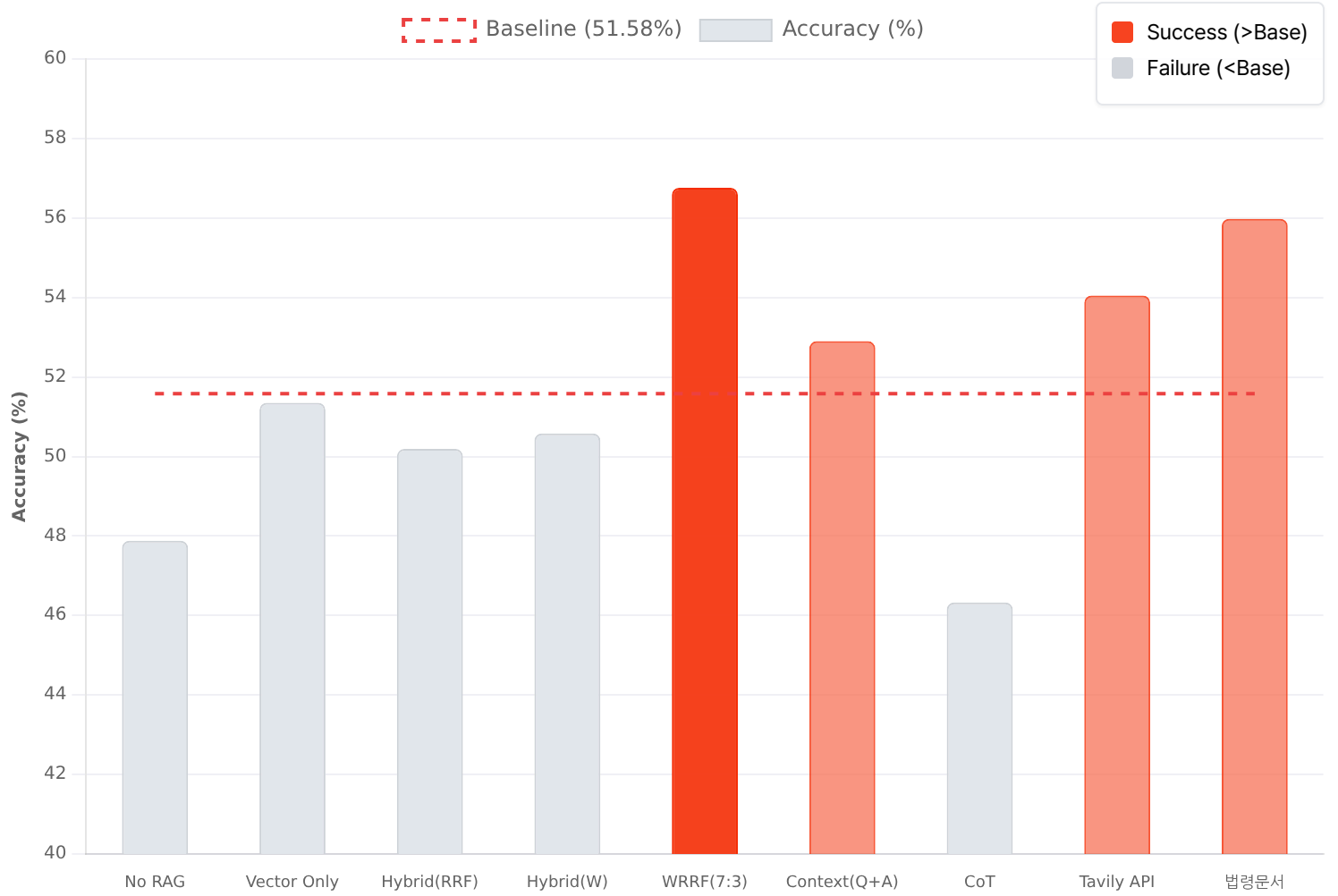
총 실험 횟수	10회
Baseline	51.58%
최고 성능	56.76%
성능 향상폭	+5.18%p

주요 결과

- **Hybrid Search + WRRF**
  가장 안정적이고 높은 성능 기록
- **CoT & 외부 검색**
  복잡도 증가 대비 성능 하락
- **법령 문서 추가**
  단순히 법령 문서를 VectorDB에 추가하였을 때 성능 하락

주요 발견점

"데이터와 검색의 기본기가 성능 향상을 이끌었으나,
CoT 및 외부 데이터 적용은
더 적합한 접근법에 대한 실험이 필요합니다."



SUCCESS STORY

성공 실험 ① Hybrid Search

검색 전략의 시행착오 끝에 찾아낸 최적의 조합 (From Vector to WRRF)

PHASE 01

Vector Only

의미 기반 검색만 사용.
법률 용어의 정확한 매칭보다
맥락적 유사도에 의존.

ACCURACY
51.35%
Baseline Level

PHASE 02

Hybrid Search 적용

단순 RRF 및 가중합(Weighted) 시도.
오히려 검색 노이즈가 증가하며
성능이 정체되거나 하락함.

ACCURACY
50.19%
↓ Performance Drop

PHASE 03 (FINAL)

WRRF 적용 및 최적화

Weighted **R**eciprocal **R**ank **F**usion 적용.
순위(Rank)와 가중치(0.7/0.3)를
동시에 고려하여 최적화.

ACCURACY
56.76%
↑ +5.18%p Boost

CORE LESSON

단순 결합이 아닌, **순위(Rank)의 안정성**과 **가중치(Weight)의 유연성**을 결합했을 때 시너지가 발생했습니다.

FAILED EXPERIMENT ANALYSIS

실패한 실험 ① Context 형식 실험

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

가설

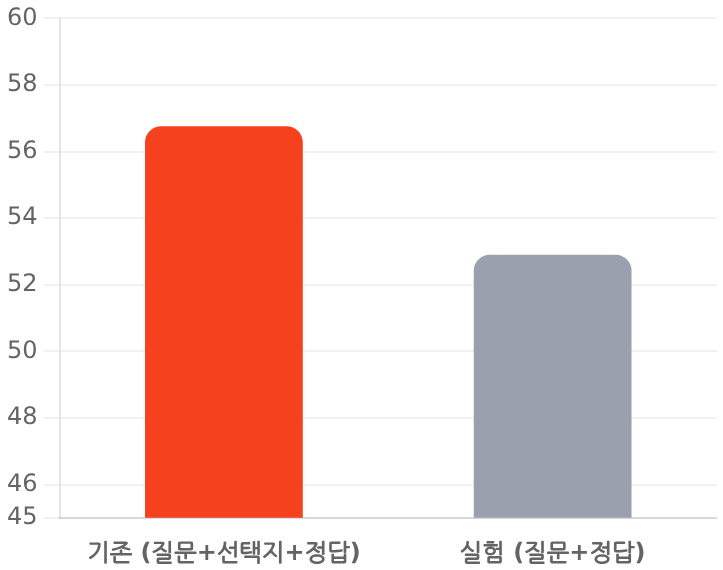
오답 선택지는
노이즈(Noise)가 아닐까?

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

실험 설정

질문 + 정답 보기만 남기고
선택지를 제거하여 DB 구축

실험 결과



↓ 3.86%p 하락

오답을 제거했더니 오히려 성능이 떨어짐

왜 실패했을까?

✕ LLM 추론 정보 부족

LLM이 정답을 판단할 때,
오답 선택지와의 **비교 정보**가
중요한 힌트로 작용함

✕ BM25 성능 저하

정답만 남기고 Vector 가중치를 높였으나
큰 효과는 없었고,
BM25 키워드 매칭 성능이 저하됨을 확인

결론

기존 형식 (질문+선택지+정답) 유지

FAILED EXPERIMENT ANALYSIS

실패한 실험 ② 외부 웹 검색

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

가설

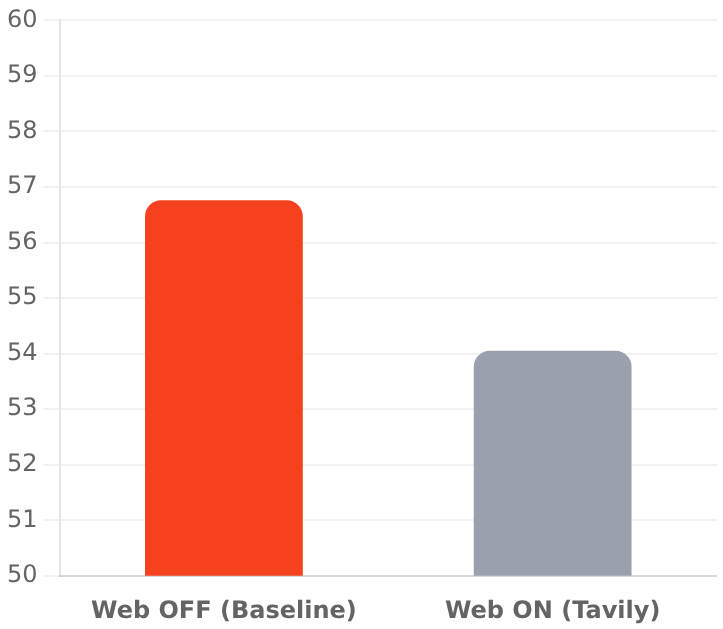
외부 검색이 부족한 지식을 채워줄 수 있지 않을까?

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

실험 설정

Tavily API 적용
기존 Context에 추가하는 방식

실험 결과



↓ 2.71%p 하락

웹 검색을 활성화했더니 성능이 오히려 떨어짐

왜 실패했을까?

✕ Context 희석

검색된 웹 정보가 중요 Context를 희석시키고 추론을 방해하는 **노이즈(Noise)**로 작용함

✕ 검색 전략 고도화 부족

단순 질문 검색이 아닌, 질문/선택지/Context를 분석해 **최적 키워드**를 추출했어야 함
(단, Latency/복잡도 증가 예상)

결론

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

가설

생각의 사슬을 만들면
정답률이 오를까?

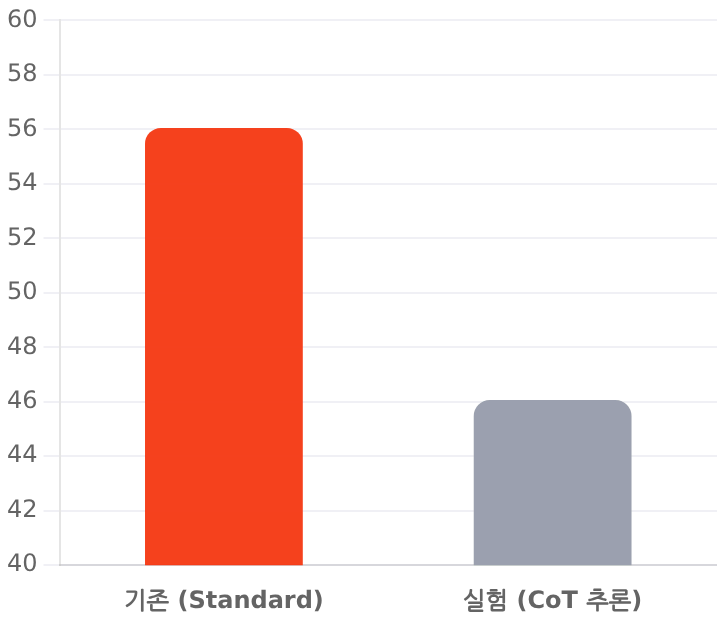
"단순히 답을 고르는 것보다,
문제 풀이를 단계적으로 따져보게 하면
문제 해결 능력이 향상될 것이다."

실험 설정

4단계 분석 유도:

1. **분석:** 문제 유형(긍정/부정) 및 핵심 법리 파악
2. **식별:** 관련 법령, 조문, 판례 탐색
3. **검토:** 각 선지(A~D)별 정오 판단
4. **결정:** 문제 유형에 맞는 최종 정답 선택

실험 결과



↓ 10.43%p 급락

예상과 달리 성능이 크게 떨어짐 (심각한 하락)

왜 실패했을까?

✕ 모델 특성 불일치

gpt-4o-mini 모델이
CoT 방식으로 **충분히 학습되지 않았거나**
경량 모델 특성상 추론 과정에서 환각 발생

✕ 도메인 특성

법률 문제는 수학/논리 문제와 달리 계산적 추론보다
정확한 판례/법령 Context의 매칭 여부
즉 정확한 정보 전달이 정답에 효과적임

결론

단계적인 추론보다
RAG의 검색 품질을 올려
Context 정확도를 높이는 것이 중요

FAILED EXPERIMENT ANALYSIS

실패한 실험 ④ 법령 문서 RAG

"실제 법령 원문을 검색하면 더 정확해지지 않을까?" → 3차례 시도 끝에 실패

가설

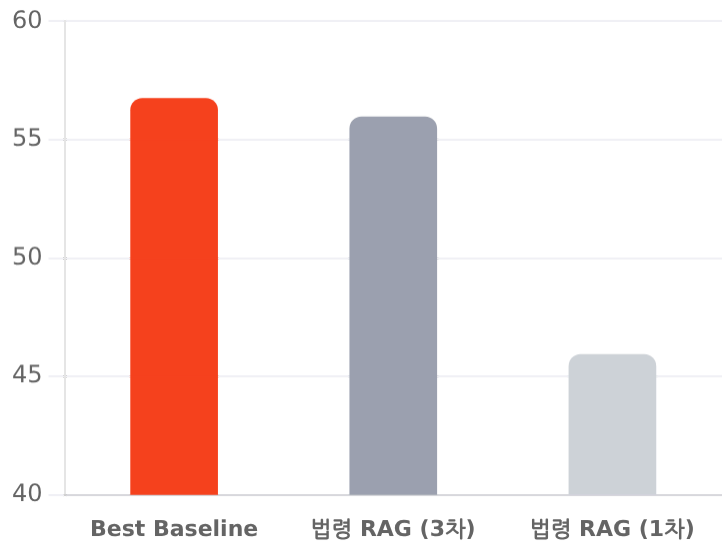
법률 문제의 정답은 법조문에 있다

"기존 Q&A 데이터셋만으로는 한계가 있다. 법령법제처의 실제 법령 문서를 추가하면 정확도가 오를 것이다."

시도 이력

- **1차** Q&A + 법령 문서 단순 통합
- **2차** 질문 유형별 DB 라우팅
- **3차** 조건부 검색 + 컬렉션 분리

실험 결과



0.78%p 부족

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

왜 실패했을까?

Context 희석

법령 문서까지 추가하여 Top-K를 늘리자,
gpt-4o-mini가 **핵심 정보를 놓치는 현상** 발생

단순 법령 정보의 한계

법령 추가로 일부 문제는 해결됐으나,
조문에 명시되지 않은 **세부 이론**이나
복합 관계 질문은 기존 검색 방식으로는 해결 불가

향후 계획

단순 텍스트 검색의 한계를 극복하기 위해,
법령 간 연관 관계를 그래프로 구조화하는
GraphRAG 도입 고려

INSIGHTS

인사이트

"방법 자체가 틀린 것이 아니라, **접근 방식의 문제**일 수 있습니다."

효과적이었던 것

- **Train Data 활용:** 학습 데이터를 Knowledge Base로 활용
- **검색 방법:** Hybrid Search + WRRF 조합
- **파라미터 튜닝:** 검색 가중치 최적화 (0.7 : 0.3)

적절치 않았던 것

- **Context 형식 변경:** 선택지 제거 시 비교 정보 부족으로 성능 하락
- **Chain-of-Thought:** sLLM 특성상 적절하지 않음
- **웹 검색(Tavily):** 단순 적용 시 현 도메인에서는 Noise로 작용, 고도화 필요
- **법령 원문 추가:** 일부 해결한 문제가 있으나,
  Context 희석 문제로 성능 저하 이후 고도화 필요

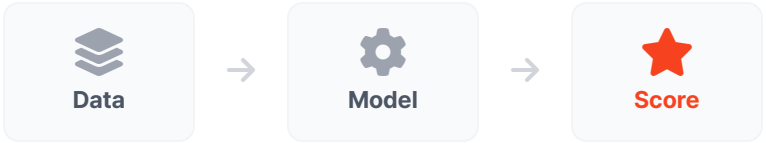
실패한 방법론(CoT, RAG 등) 자체가 틀린 것이 아닐 수 있다.
하지만 짧은 기간에서는 단순한 구조를 바탕으로,
기본적인 RAG 파이프라인을 구축하고 조정하는 것이 성능 향상을 이끌어냈습니다.

FUTURE ROADMAP

추가 개선 방향

시간이 더 있었다면 시도해보고 싶었던
고도화 전략 2가지를 제안합니다.

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



지식 부족 해결

Knowledge Base 확장

현재 문제: **sLLM에 내재된 세부 법률 개념과 최신 판례 지식 부족**

개선 방법:

- 법률 자격증 이론서/요약본을 **Q&A화**하여 RAG 통합
- 신뢰성 확보를 위한 출처/버전 메타데이터 관리

기대효과: 전문 법률 지식을 보강하여 답변의 깊이와 정확도 향상

복합 추론 강화

GraphRAG 도입

현재 문제: **단순 검색으로는 여러 법률 간 복합 관계 문제의 정확도가 낮음**

개선 방법:

- 법령 문서를 기반으로 **법률 간 관계 그래프** 구축
- Multi-hop 검색 및 논리적 근거 반환

기대효과: 법조문 간 연결성을 파악하여 고차원적인 법률 추론 가능

SUMMARY & CONCLUSION

마무리 요약

01. 과제 해석

엔지니어링 사고력 평가

"sLLM기반 Agent 구조의
시스템 성능 개선 능력과
실무 요구사항 대응 역량을 평가"

02. 접근 방식

가설 - 실험 - 검증

가설을 기반한 반본적인 실험을 통해
Baseline 이상의 성능을 달성한 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"

실패한 방법론(CoT, RAG 등) 자체가 틀린 것이 아닐 수 있다.
하지만 짧은 기간에서는 단순한 구조를 바탕으로,
기본적인 RAG 파이프라인을 구축하고 조정하는 것이 성능 향상을 이끌어냈습니다.

APPENDIX

예상 Q&A

면접관님의 질문에 대비하여 준비한 예상 질의응답입니다.

Q1 RRF, Weighted, WRRF의 차이는?

WRRF는 RRF(순위 기반)의 안정성과 Weighted(점수 가중치)의 장점을 결합한 방식입니다.
순위 정보와 모델별 중요도를 모두 반영하여 검색 성능을 극대화합니다.

Q2 CoT가 성능을 악화시킨 이유는?

sLLM(gpt-4o-mini)은 복잡한 CoT 추론 학습이 되어 있지 않아,
프롬프트만으로는 효과가 제한적입니다.
정확한 Context를 제공하는 RAG 품질 향상이 핵심입니다.

Q3 현재 시스템의 한계 극복 방안은?

지식 부족 문제는 전문 서적 기반의 KB 확장으로,
복합 추론 문제는 법률 간 관계를 그래프로 모델링하는
GraphRAG 도입으로 해결할 수 있습니다.

Q4 WRRF 가중치 0.7:0.3의 근거는?

다양한 가중치 조합(0.5:0.5, 0.6:0.4 등)을 실험한 결과,
Vector 검색에 0.7, Keyword 검색에 0.3을 부여했을 때
가장 높은 성능(56.76%)을 기록했습니다.

Q5 실패한 실험에서 얻은 교훈은?

특정 방법론(Web Search, CoT) 자체가 틀린 것이 아니라,
sLLM 환경과 법률 도메인 특성에 맞는
구체적인 적용 방식(Engineering)이 중요함을 확인했습니다.

Q6 train.csv를 Knowledge Base로 쓴 이유는?

데이터 분석 결과 train과 dev 셋 간에 중복 및 유사 문항 패턴이 확인되었으며,
과제의 의도가 이를 RAG 소스로 활용하여 성능을 최적화하는 것이라 판단했습니다.