Building and Evaluating a Legal-Domain RAG system: A Comparative Study of Retrievers and Rerankers in the Real Estate Tax field

Seong-Yeoup Jeong[1], Yoo-Bin Park[2], Youn-Ha Kang[3], Seong-Soo Park[4], Byeong-Seon Park†[5]

1 Division of Data Science, Suwon University
2, 3 School of Information and Communication Engineering, Chung Buk National University
4 Dept. of Business Administration, Tech University of Korea
5† SK Inc. C&C, Corresponding Author

unknownlimitless0301@gmail.com, **yubin11890@gmail.com**, **kyh030314@gmail.com**, **asdzxc46093838@gmail.com**, **parkbscorp@gmail.com**

요약

본 연구는 법률 도메인, 그 중에서도 부동산세 데이터를 기반으로 Retrieval-Augmented Generation(RAG) 시스템을 구축하고, RAG 시스템을 구성하는 구성요소들이 답변 성능에 미치는 영향 및 성능 지표를 정량적으로 비교하였다.

본 연구를 통해 도출된 자료는 부동산세 자문 상황에 적합한 RAG 시스템 설계에 있어서 RAG 구성요소들의 선택을 돕고, 향후 LangGraph 기반의 AI Agent 구현의 경험적 근거가 될 수 있을 것이다.

1. 서론

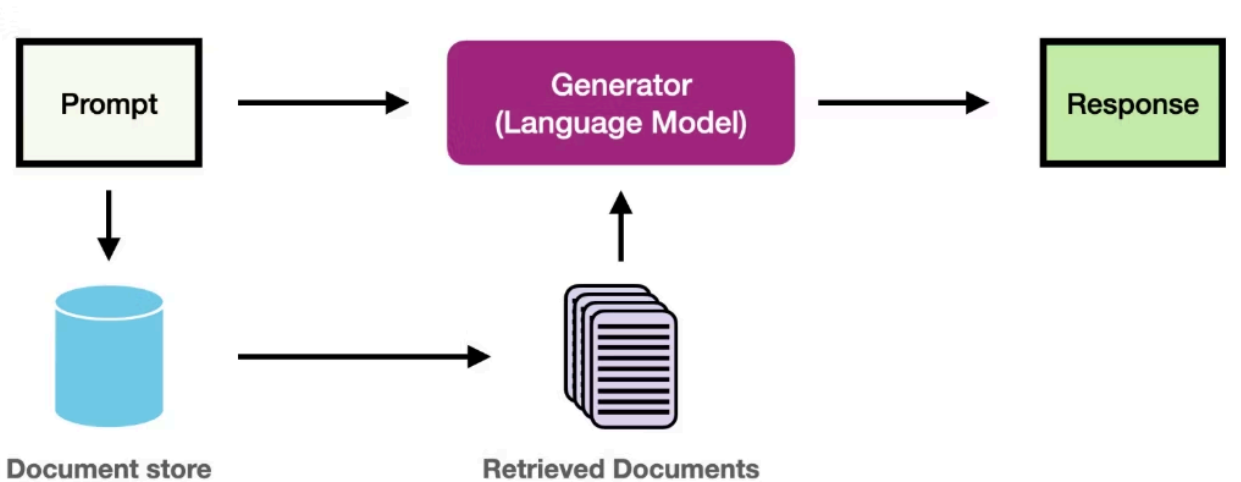
최근 몇 년 사이 거대 언어 모델(Large Language Model, LLM)은 단순한 문장 생성을 넘어서 기업의 업무 프로세스를 자동화하거나 연구 활동을 보조하는 등 인간의 도구로 급속히 자리잡고 있다. 그러나 현재 상용화된 LLM들은 광범위한 주제에 대한 데이터로 학습되어 있어, 특정 도메인에 특화된 답변을 받기에는 한계가 명확하다. 또한 이들 모델들은 학습한 데이터의 시점, 그리고 모델 파라미터에 의존하기 때문에 최신 정보에 대한 답변이 불명확하고 거짓된 답변을 제공하는 할루시네이션(Hallucination) 현상이 빈번하게 발생하는 문제가 있다[1].

이를 해결하기 위한 방법으로 LLM에 외부 지식을 검색·결합하는 RAG(Retrieval-Augmented Generation) 구조가 제안되었다[2]. 이 방법은 LLM 모델 자체를 재 학습시키지 않고도 고품질의 답변을 얻을 수 있다는 점에서 각광받는 기법으로 쓰이고 있다.

한편 현재 여러 도메인에 대해 RAG 연구는 활발해지고 있으나, 최근 연구들은 대체로 심리학 및 인지과학 도메인에서 RAG 연구가 이루어지고 있고, 법률 도메인에서의 RAG 연구는 상대적으로 적은 편이다. 또한 연구들은 RAG의 핵심적 구성 요소들인 리트리버(Retriever)와 리랭커(Reranker) 중 리트리버들의 비교를 위주로 진행되고 있고, 평가 지표, LLM 설정을 동일 조건으로 변인 통제한 상태에서 리트리버들과 리랭커들의 성능을 개별적으로 비교한 연구는 드문 편이다.

본 연구는 이 점에 주목하여 부동산세 분야를 대상으로 임베딩 모델과 LLM 설정을 고정(변인 통제)한 상태에서 리트리버들과 리랭커들의 성능을 각각 RAGAS framework[3]으로 정량적으로 평가함으로써, 법률 도메인에서 RAG 설계에 실험적 근거를 제공하고자 한다.

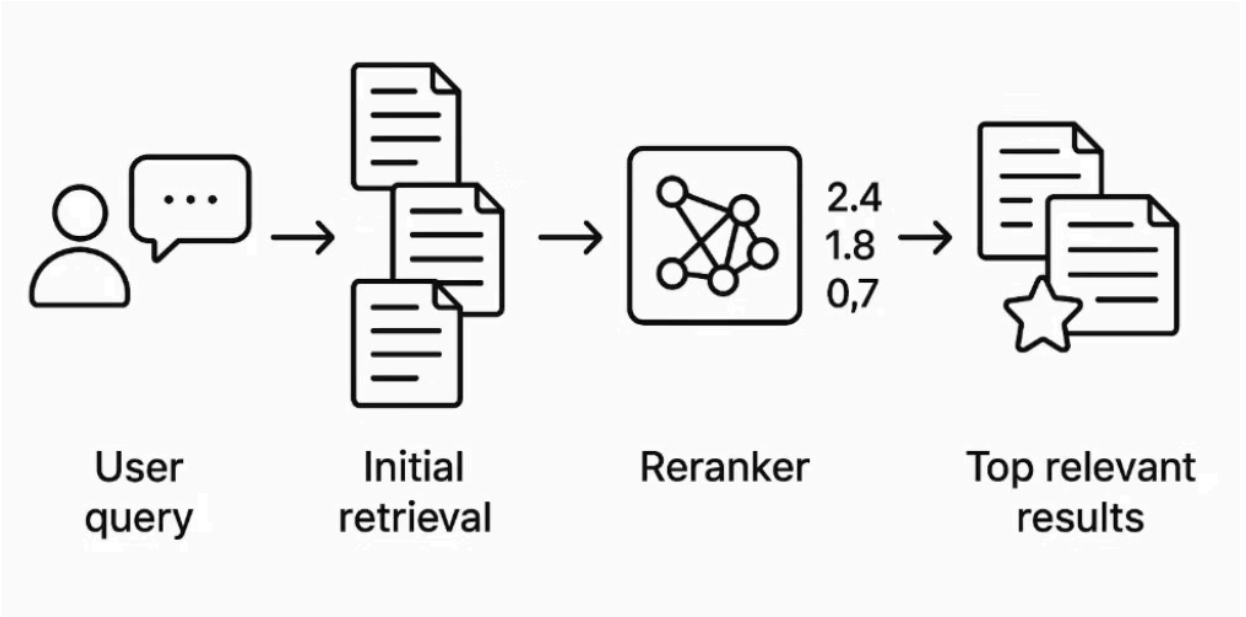
RAG(Naive RAG)



LLM의 환각(Hallucination) 현상을 해결하기 위한 방법론 중 하나

- 사용자의 질문을 받으면 Vector DB 또는 Vector Store(Document Store)에서 Retriever가 질의와 유사한 문서들을 search 함
- 이렇게 search 된 Retrieved Documents가 LLM에 주입되어 답변이 제공되는 구조

Advanced RAG(본 실험에서 적용)



위의 RAG 구조도에서 **'Retrieved Documents'** 단계 이후에 **Reranker가 삽입**되어 **search된 문서들 중 사용자의 질의와 유사한 순서로 질문을 재순위화** 하여 Top relevant Results를 제공하고, 이를 바탕으로 LLM이 답변을 생성하는 구조

2. 실험

2.1 RAG 시스템 구축 과정

1. 국가법령정보공동활용 API에서 종합부동산세 판례 1,268건을 수집하고 본문 텍스트를 추출·정규화
2. 문단/조문 경계를 최대한 보존하기 위해 규칙 기반 문자 청킹을 적용하였으며, chunk_size=900, chunk_overlap=150으로 고정
3. 임베딩은 intfloat/multilingual-e5-large-instruct 모델로 생성하였고, 임베딩 단계에서 모델 입력이 512 토큰을 초과하면 절단(truncation) 하도록 하여 임베딩 데이터를 제작

2.2 RAGAS 프레임워크와 평가를 위한 데이터셋 구성

RAGAS는 RAG 시스템의 성능을 측정하도록 하는 open-source framework으로, RAG 시스템에 바로 탑재하여 쉽게 평가가 가능하다.

본 실험에서는 수집한 판례 데이터에서 구성한 50개의 Q-A 쌍을 정답 데이터로 제작하여, RAG 구성요소들의 성능을 측정하였다.

평가 데이터

50개 Q-A 쌍

판례 기반 정답 데이터(수동으로 생성)

2.3 리트리버 성능 비교 분석

수집된 판례 데이터를 기반으로 RAG 시스템의 핵심 구성 요소인 리트리버의 성능을 RAGAS 프레임워크를 활용해 비교 분석하였다.

Dense Retriever

사전 학습된 신경망 모델을 이용해 질의와 문서를 고차원 벡터 공간에 임베딩하고, 의미적 유사도를 기반으로 문서를 검색한다. 키워드 일치보다 **의미적 유사성**에 중점을 두어 **다양한 표현이나 동의어 처리**에 효과적이다.

TF-IDF Retriever

문서 내 특정 단어의 중요도를 통계적으로 계산하여 문서를 검색하는 전통적인 기법이다. 질의와 문서 간의 **정확한 키워드 매칭**에 강하다.

BM25 Retriever

TF-IDF를 개선한 확률적 랭킹 함수로, **단어 빈도 포화** 완화 및 **문서 길이 정규화**를 적용해 더 정교한 관련성 점수를 산출한다. TF-IDF보다 발전된 키워드 기반 검색을 제공하며 **특정 키워드의 중요도가 높은 상황에서의** 검색에 유용하다.

실험 설정 및 방법론:

리트리버 비교 실험은 부동산세에 대한 실제 판례 데이터를 기반으로 구성된 50개의 질의-응답(Q-A) 쌍을 평가 데이터셋으로 활용하였다. 검색된 문서는 LLM(**GPT-4o, temperature 0**)의 입력으로 주어져 응답을 생성했으며, RAGAS 프레임워크의 지표들을 측정했습니다. 벡터 스토어는 **Naive Vector Store**를 사용했으며, 검색 설정은 리트리버 간 공정성을 위해 통일했습니다. Context Precision 값은 소수점 넷째 자리에서 반올림하여 제시하였다.

리트리버 성능 비교 결과

(평가 데이터: 50개 판례 기반 Q-A 쌍, Naive Vector Store, LLM=GPT-4o(temperature=0) 적용)

리트리버	Context Precision	Context Recall	Overall Score
Dense	0.993	0.91	0.967556
TF-IDF	0.881	0.86	0.860875
BM25	0.891	0.92	0.901972

실험 결과 분석:

평가 결과, **Context Precision** 측면에서는 Dense Retriever가 0.993으로 압도적으로 높은 성능을 보였다. 이는 Dense Retriever가 질의와 관련된 **정확한 문맥(context)**을 검색하는 능력이 탁월함을 의미하며, LLM이 불필요하거나 잘못된 정보를 기반으로 응답을 생성할 위험을 크게 줄여준다고 볼 수 있을 것이다. 반면 TF-IDF와 BM25는 상대적으로 낮은 Context Precision을 기록하여, 키워드 기반 검색의 한계로 인해 관련성이 낮은 문서가 포함될 가능성이 있음을 시사한다다.

Context Recall에서는 BM25 Retriever가 0.92로 가장 우위를 보였다. Context Recall은 관련성 있는 모든 문맥 중 얼마나 많은 부분을 실제로 검색했는지를 나타내는 지표로, BM25가 키워드 매칭과 문서 길이 정규화를 통해 넓은 범위의 관련 문서를 효과적으로 찾아냈음을 알 수 있다.

최종 성능 지표인 **Overall Score**에서는 Dense Retriever가 0.967556으로 가장 우수한 결과를 보였다. 이는 높은 Context Precision과 준수한 Context Recall이 결합된 결과로, 법률 질의와 같이 다양한 표현 변형이 존재하고 미묘한 의미 차이가 중요한 도메인에서 **의미론적 정보를 정확하게 포착하는 Dense Retriever**의 강점이 전체적인 RAG 시스템 성능 향상에 크게 기여했음을 시사한다.

결론적으로, 본 실험은 법률 분야의 RAG 시스템 구축 시 단순 키워드 매칭 방식보다는 질의의 **심층적인 의미를 이해하고 정확한 문맥을 제공하는 Dense Retriever**의 활용이 전반적인 응답 품질을 높이는 데 매우 효과적임을 입증한다. 향후 연구에서는 Dense Retriever의 임베딩 모델 최적화와 함께, 각 리트리버의 장점을 결합하는 하이브리드 검색 방식에 대한 탐색이 필요할 것이다.

2.4 리랭커 비교

리랭커 비교는 Dense Retriever를 모두 동일하게 고정하고 다섯 종의 리랭커들을 RAG 시스템으로 구현하여 RAGAS 평가를 수행하였다. 평가는 표 2와 같이 RAGAS 평가 지표 중 *Context Precision, Context Recall, Faithfulness*를 중점적으로 사용하였고, 이들 지표를 동등 가중으로 합성한 *Overall Score*를 보조적으로 제시한다.

비고: 본 비교에서는 Answer Relevancy 를 보고하지 않았다. 본 연구의 리랭커 비교는 Retrieve, Rerank 단계에서의 품질에 초점을 두었고, 해당 지표는 본 설계에서 리랭커 민감도 가 낮아 해석력이 제한적이었다.

구현한 리랭커들

BM25 Reranker

Dense Retriever가 검색한 문서들을 **BM25** 알고리즘에 따라 리랭킹하여 구현.

파이프라인 흐름

```
사용자 질문
    ↓
① Dense Retriever (E5 임베딩 + 코사인 유사도)
    → 100개 후보 문서 검색
    ↓
② BM25 Reranker (어휘 기반 매칭)
    → 토큰 중첩도 기반으로 재정렬
    → 상위 12개 선택
    ↓
③ LLM (GPT-4o)
    → 최종 답변 생성
```

Cohere Reranker

Cohere에서 제공하는 **rerank-multilingual-v3.0** 모델로 리랭킹 로직을 구현

Hybrid Reranker

파이프라인 흐름

```
사용자 질문
    ↓
① Dense Retriever (E5 임베딩 + 코사인 유사도)
    → 100개 후보 문서 검색
    ↓
② Hybrid Reranker
    ├─ BM25 점수 계산 (키워드 매칭)
    ├─ E5 임베딩 점수 계산 (의미 유사도)
    ├─ Min-Max 정규화
    └─ CombSum 합산 → 상위 12개 선택
    ↓
③ LLM (GPT-4o)
    → 최종 답변 생성
```

LLM Reranker

파이프라인 흐름

```
사용자 질문
    ↓
① Dense Retriever (E5 임베딩 + 코사인 유사도)
    → 50개 후보 문서 검색
    ↓
② LLM Reranker (GPT-4o)
    ├─ 각 문서마다 LLM API 호출 (50번 호출)
    ├─ 질문-문서 관련성 평가 (0-10점)
    └─ 점수 기반 재정렬 → 상위 10개 선택
    ↓
③ LLM (GPT-4o)
    → 최종 답변 생성
```

Rule Reranker

- '조문', '항', '호' 패턴에 따라 가중치를 부여하여 리랭킹.
- 사실상 하드코딩이라 볼수 있다.

파이프라인 흐름(예시)

```
사용자 질문: "종합부동산세법 제23조의 내용은?"
    ↓
① Dense Retriever (E5 임베딩)
    → 80개 후보 문서 검색
    ↓
② Rule-Based Boost Reranker
    ├─ 각 문서의 임베딩 유사도 계산 (기본 점수)
    ├─ 법률 규칙 부스트 계산
    │    ├─ "제23조" 포함 → +0.3
    │    ├─ "제1항" 포함 → +0.2
    │    ├─ "종합부동산세" 포함 → +0.1
    │    └─ "제23조" (질문 키워드) → +0.05
    ├─ 기본 점수 × (1 + 부스트)
    └─ 재정렬 → 상위 12개 선택
    ↓
③ LLM (GPT-4o)
    → 최종 답변 생성
```

리랭커 성능 비교 결과

(Context Precision 값은 소수점 넷째 자리에서 반올림, 평가 데이터: 50개 판례 기반 Q-A 쌍, Naive Vector Store, LLM=GPT-4o(temperature=0) 적용)

리랭커	Context Precision	Context Recall	Faithfulness	Overall Score
BM25	0.621	1	0.33	0.709276
Cohere	0.975	1	1	0.964421
Hybrid	0.977	1	0	0.714807
LLM	0.643	1	1	0.881360
Rule	0.557	1	1	0.859829

주*: BM25: BM25* 알고리즘으로 리랭킹*. Cohere: rerank-multilingual-v3.0* 모델로 리랭킹*. Hybrid: BM25* 점수와 임베딩 유사도를 정규화 후 합산*. LLM: GPT-4o* 모델이 질문*-문서 관련성을 0~10 사이의 점수로 평가하도록 구현한 pointwise 리랭킹(temperature=0). Rule: '조문', '항', '호'* 패턴 및 법률 키워드에 따라 가중치를 부여하여 리랭킹*.*

리랭커 실험 결과 분석

실험 결과, Cohere에서 제공하는 rerank-multilingual-v3.0 리랭커를 사용한 경우가 전반적으로 가장 우수한 성능을 보였다.

Hybrid Reranker는 *Context Precision*은 가장 높았으나(0.977), *Faithfulness*가 0으로 현저하게 낮았다. 이러한 특성은 답변 오류를 유발할 위험이 높으므로 사용에 주의가 필요하다.

비고

본 비교에서는 *Answer Relevancy*를 보고하지 않았다. 본 연구의 리랭커 비교는 *Retrieve, Rerank* 단계에서의 품질에 초점을 두었고, 해당 지표는 본 설계에서 리랭커 민감도가 낮아 해석력이 제한적이었다.

3. 한계 및 논의

본 연구는 법률 도메인(부동산세)에서 RAG 구성요소를 변인 통제 하에 비교하였다는 점에 의의가 있으나, 몇 가지 한계가 있다.

일반화 가능성

리트리버와 리랭커 평가 시 단일 임베딩 모델과 단일 Vector Store를 사용하여 일반화 가능성을 보수적으로 해석해야 한다.

문서 분할 이슈

512 토큰 단위로 끊어 임베딩 하였기에 일부 문서들은 문단/조문 경계가 제대로 나뉘어지지 않았을 가능성이 있다.

효율성 분석 제한

이번 비교는 정확도를 중심으로 비교하여 응답이 걸리는 시간을 포함한 효율성 분석은 제한적이었다.

결과적으로, 법률 질의에서는 `Dense Retriever + Cohere Reranker` 조합이 가장 일관된 이점을 보였다. Dense Retriever + LLM Reranker 조합은 *Context Precision*은 중간이나, Overall Score 약 0.88을 기록하여, 품질-비용을 고려할 때 선택 가능한 대안으로 해석된다.

4. 결론

본 연구에서는 부동산세 판례 데이터를 기반으로 동일한 임베딩 모델과 동일한 LLM 설정을 기반으로 리트리버 3종, 리랭커 5종을 비교하여 RAG의 검색·재순위 성능을 RAGAS로 평가하였다.

리트리버 결과

Dense Retriever가 전반적으로 가장 우수한 성능을 보였다.

- Overall Score: 0.967556
- Context Precision: 0.993
- 의미론적 검색의 강점 확인

리랭커 결과

Cohere Reranker가 가장 우수한 성능을 보였다.

- Overall Score: 0.964421
- 모든 지표에서 균형잡힌 성능
- 법률 도메인 적합성 입증

향후 연구에서는 응답 지연 시간 및 API 호출에 따른 효율성을 분석함과 동시에, LangGraph 기반 다단계 AI Agent로 RAG 파이프라인을 고도화하여 RAGAS 지표와 실사용 안정성을 함께 평가할 예정이다.

ACKNOWLEDGEMENT

※ 본 논문은 과학기술정보통신부 대학디지털교육역량강화 사업의 지원을 통해 수행한 한이음 드림업 프로젝트 결과물입니다.

참고문헌

1. Ji, Z et al., "Survey of Hallucination in Natural Language Generation," ACM Computing Surveys, 55(12), 248:1–248:38, 2023.
2. Lewis, P. et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," NeurIPS 2020, Vancouver, Canada, 2020, pp. 9459–9474.
3. Es, S. et al., "Ragas: Automated Evaluation of Retrieval Augmented Generation," arXiv preprint arXiv:2309.15217, 2023.