법률 도메인 RAG 시스템 구축과 성능 평가: 부동산세에서의 리트리버·리랭커 비교 연구

정성엽 [1], 박유빈 [2], 강윤하 [3], 박성수 [4], 박병선 [5]
[1] 수원대학교 데이터과학부 학부생
[2, 3] 충북대학교 정보통신공학부 학부생
[4] 한국공학대학교 경영학부 학부생
[5]SK Inc. C&C

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

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

요 약

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 LLM 들은 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 LLM 에 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 RAG 의 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.

2. 실험

2.1 RAG 시스템 구축 과정

국가법령정보공동활용 API 에서 종합부동산세 판례 1,268 건을 수집하고 본문 텍스트를 추출·정규화하였다. 문단/조문 경계를 최대한 보존하기 위해 규칙 기반 문자 청킹을 적용

하였으며, chunk_size=900, chunk_overlap=150 으로 고정하였다. 임베딩은 intfloat/multilingual-e5-large-instruct 모델로 생성하였고, 임베딩 단계에서 모델 입력이 512 토큰을 초과하면 절단(truncation) 하도록 하여 임베딩 데이터를 제작하였다.

2.2 RAGAS 프레임워크와 평가를 위한 데이터셋 구성

RAGAS 는 RAG 시스템의 성능을 측정하도록 하는 open-source framework 으로, RAG 시스템에 바로 탑재하여 쉽게 평가가 가능하다. 본 실험에서는 수집한 판례 데이터에서 구성한 50 개의 Q-A 쌍을 정답 데이터로 제작하여, RAG 구성요소들의 성능을 측정하였다.

2.3 리트리버 비교

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

<표 1> 리트리버 성능 비교 (*Context Precision* 값은 소수점 넷째 자리에서 반올림, Naive Vector Store, LLM=GPT-4o(temperature=0) 적용)

세 종류의 리트리버들을 RAG 로 구현하고 RAGAS 평가를 수행하였으며, 그 결과는 표 1 과 같았다. *Context Recall* 측면에서는 BM25 Retriever 가 가장 우위를 보였으며, 최종 성능인 *Overall Score* 는 Dense Retriever 가 가장 우수하였다. 이는 법률 질의처럼 표현 변형이 많은 환경에서 의미론적 정보를 잘 포착하는 Dense Retriever 가 전반적 성능에서 유리함을 시사한다.

2.4 리랭커 비교

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

<표 2> 리랭커 성능 비교 (*Context Precision* 값은 소수점 넷째 자리에서 반올림, Naive Vector Store, LLM=GPT-4o(temperature=0) 적용)

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 비교는 Dense Retriever 를 모두 동일하게 고정하고 다섯 종의 리랭커들을 RAG 시스템으로 구현하여 RAGAS 평가를 수행하였다. 평가는 표 2 와 같이 RAGAS 평가 지표 중 *Context Precision*, *Context Recall*, *Faithfulness* 를 중점적으로 사용하였고, 이들 지표를 동등 가중으로 합성한 *Overall Score* 를 보조적으로 제시한다. 실험 결과, Cohere 에서 제공하는 rerank-multilingual-v3.0 리랭커를 사용한 경우가 전반적으로 가장 우수한 성능을 보였다. Hybrid Reranker 는 *Context Precision* 은 가장 높았으나(0.977), *Faithfulness* 가 0 으로 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. 첫째, 리트리버와 리랭커 평가 시 단일 임베딩 모델과 단일 Vector Store 를 사용하여 일반화 가능성을 보수적으로 해석해야 한다. 둘째, 512 토큰 단위로 끊어 임베딩 하였기에 일부 문서들은 문단/조문 경계가 제대로 나뉘어지지 않았을 가능성이 있다. 셋째, 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 0.88 을 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 3 종, 리랭커 5 종을 비교하여 RAG 의 검색·재순위 성능을 RAGAS 로 평가하였다. 그 결과, 리트리버에서는 Dense Retriever 가 전반적으로 가장 우수하였고, 리랭커에서는 Cohere Reranker 가 가장 우수한 성능을 보였다. 향후 연구에서는 응답 지연 시간 및 API 호출에 따른 효율성을 분석함과 동시에, LangGraph 기반 다단계 AI Agent 로 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.