AutoRAG를 이용한 금융 문서에 가장 최적화된 RAG 시스템 구현에 관한 연구

임재훈[1], 서장원[2]
[1]동서울대학교 컴퓨터소프트웨어학과 학부생
[2]동서울대학교 컴퓨터소프트웨어학과 교수
pungddang@naver.com, jwsuh@du.ac.kr

A Study on implementing the most optimized RAG system for financial document using AutoRAG

Jae-Hoon Lim[1], Jang-Won Suh[2]
[1]Dept. of Computer Software, Dongseoul University
[2]Dept. of Computer Software, Dongseoul University

요 약

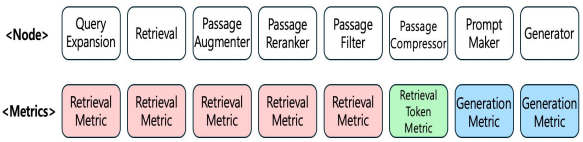
대규모 언어 모델(LLM)은 많은 영역에서 우수한 성능을 발휘한다. 하지만 오래된 지식, 환각, 불투명한 추론 프로세스 등의 문제가 존재한다. 이러한 문제의 해결책으로 외부 데이터베이스의 지식을 통합하는 Retrieval-Augmented Generation(RAG)시스템이 떠오르고 있다. 그러나 RAG 시스템을 구현하기 위한 파이프라인의 구성은 복잡하며 각 시나리오에 대한 실험과 평가는 시간이 걸리고 번거롭다. 이 논문에서는 일련의 실험과 평가를 한 번에 실험하고 각 단계별 평가 점수를 측정할 수 있는 AutoRAG라는 프레임워크를 사용하여 금융 문서에 가장 최적화된 RAG 파이프라인을 탐색하여, Naive RAG와 Advanced RAG 시스템을 비교하였다. 두 경우 모두 F1점수, NDCG점수, mAP점수를 사용하여 평가하였다. 최종 성능지표의 결과로 Advanced RAG 시스템이 F1, NDCG, mAP에서 각각 0.062, 0.168, 0.106 만큼 우수한 것으로 나타났다.

1. 서론

LLM은 질문을 이해하고 유창한 언어 텍스트를 생성하는 인상적인 능력을 보이고 있다. 하지만 오래되거나 잘못된 지식으로 인해 심한 환각 현상(Hallucination)이 발생할 수 있다. 이는 대부분의 실제 애플리케이션에서 사용자 경험에 영향을 끼칠 수 있다[1]. 환각 현상을 해소하기 위해 RAG 시스템을 사용하게 되었다. 그러나 RAG 시스템을 구현하는 것은 복잡하며 데이터, 사용 사례, 복잡한 설계 결정에 대한 깊은 이해가 필요하며, 이러한 시스템의 평가에는 어려움이 따르고 다각적인 방식으로 검색 정확도와 생성 품질의 평가가 필요하다[2]. 그래서 본 논문에서는 위의 복잡한 작업들을 보다 간소화 해주는 AutoRAG라는 프레임워크(Framework)를 사용하여 금융관련 상품 설명서에 가장 최적화된 RAG 시스템 구현에 대한 연구를 진행하고자 한다.

2. AutoRAG(Auto Retrieval-Augmented Generation)

많은 RAG 시스템과 모듈이 존재하지만, 어떤 RAG 시스템 구조가 자신의 데이터와 사용 사례에 적합한지 알아내는 것은 어렵다. 이러한 문제를 손쉽게 해결하기 위해 AutoRAG[3] 프레임워크가 등장하게 되었다. 본 논문에서는 보다 최적화된 시스템을 찾기 위한 방법으로 그림1의 기존 Retrieval, Prompt maker, Generator 노드의 Naive RAG 시스템에서 Query expansion, Passage 노드를 추가로 적용한 Advanced RAG를 사용하여 질문을 더욱 잘 이해하고, 검색된 문서를 재정렬하여 더욱 정확한 답변을 도출할 수 있는 Advanced RAG 방법론을 적용하여 검색기의 탐색 과정을 개선하고자 하였다.



(그림 1) AutoRAG의 구성

3. 실험 환경

<표1>은 AutoRAG를 사용하여 실험을 진행한 컴퓨터의 하드웨어 및 소프트웨어 규격이다. <표2>는 본 논문에 사용된 데이터셋인 19페이지 정도의 PDF 파일의 문서인 KB 국민은행의 청년도약플러스적금 상품의 설명서에 가장 최적화된 Advanced RAG 시

스템을 탐색하기 위해 실험을 수행할 모듈의 규격이다.

<표 1> 컴퓨터의 하드웨어, 소프트웨어 규격

구분	항목	내용
H/W	CPU	M2 Pro -12core
	GPU	M2 Pro -19core
	RAM	32GB
S/W	OS	macOS 14.6.1
	Library(with version)	python 3.11.9
		autorag 0.2.12

<표 2> RAG 시스템의 최적화 탐색을 위한 모듈 규격

metric	node type	module	optional settings
Retrieval metric	Query expansion	pass_query expansion	no optional settings
		query decompose	llama_index_llm, openAI's gpt-4o-mini, temperature:0.0~1.0
		HyDE	llama_index_llm, openAI's gpt-4o-mini, max_token: 64
	Retriever	BM25	default tokenizer from AutoRAG
		VectorDB	openAI's Embedding_model
		Hybrid RRF	weight_range(3, 5)
		Hybrid CC	normalization_method; mm, tmm, z, dbsf
	Passage reranker	pass_reranker	no optional settings
		Koreranker	no optional settings
		UPR	no optional settings
		TART	no optional settings
		Cohere	api_key, batch: 64, model: rerank-multilingual-v2.0
Generation metric	Prompt maker	fstring	2 prompts were used to compare to each other
		Long context reorder	2 prompts were used to compare to each other
	Generator	openai(LLM)	openAI's gpt-4o-mini, temperature: 0.1 ~ 1.0

4. 실험 결과

Naive RAG 시스템의 검색 성능 결과인 <표3>과 Advanced RAG 시스템의 검색 성능 결과인 <표4>에서는 F1, NDCG, mAP의 성능지표[4]를 사용하였다.

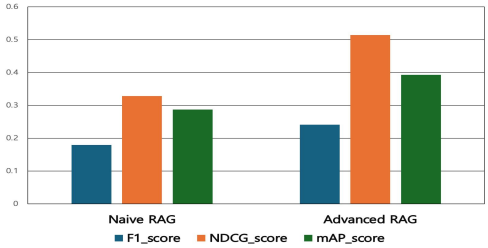
<표 3> Naive RAG 시스템 실험 결과

metric	node type	module	결과치	
Retrieval metric	Retriever	VectorDB	F1	0.179
			NDCG	0.328
			mAP	0.287

<표 4> Advanced RAG 시스템 실험 결과

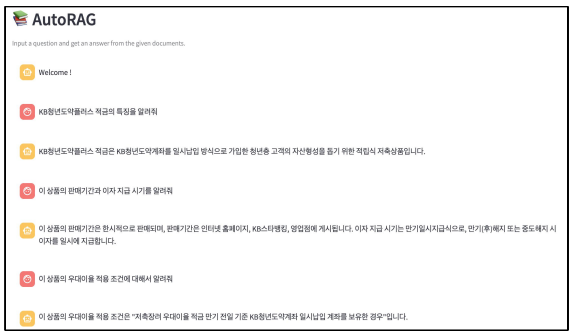
metric	node type	module	결과치	
Retrieval metric	Passage reranker	Koreranker with BM25 retriever	F1	0.241
			NDCG	0.514
			mAP	0.393

그림 5는 Naive RAG 시스템과 Advanced RAG 시스템의 Retrieval metric의 성능 점수를 그래프로 시각화한 것이다.



(그림 2) Naive RAG와 Advanced RAG 시스템의 결과치 비교 그래프

그림 3은 최적화 RAG 시스템을 Streamlit 웹을 사용하여 LLM에게 직접 훈련된 데이터에 대해서 질문을 입력하면 그에 대한 답을 출력하는 화면이다.



(그림 3) 최적화 RAG 시스템의 웹 구현 결과 화면

4. 결론

AutoRAG를 활용함으로써 각 모듈별 최적의 모듈을 찾아, 데이터에 적합한 RAG 시스템을 구현하였고 그 성능을 실험을 통해 증명하였다. 실험결과, Advanced RAG가 Naive RAG 시스템에 비해 F1은 0.062, NDCG는 0.186, mAP는 0.106 만큼 결과가 향상되었다. 앞으로 AutoRAG는 다양한 모듈의 지원으로 인해 더욱 실험이 간편해지고, 구현의 효율성이 높아질 것이다. 향후 음성인식/합성기술을 적용하여 음성만을 사용하여 동작하는 RAG 시스템 최적화에 대해 연구하고자 한다.

참고문헌

[1] Shi-Qi Yan et.al "Corrective Retrieval Augmented Generation", arXiv:2401.15884, 2024

[2] Daniel Fleischer et.al "RAG Foundry: A Framework for Enhancing LLMs for Retrieval Augmented Generation" arXiv:2408.02545, 2024

[3] https://docs.auto-rag.com/structure.html

[4] https://docs.auto-rag.com/evaluate_metrics/retrieval.html