RAG 기반 LLM 성능 평가 및 검증을 위한 LangChain 활용 RAGA 방법론 연구[1]

정효정, 송주현, 서상훈, 임진효, 이현상, 김동균
경북대학교 컴퓨터학부 글로벌소프트웨어융합전공, 심화컴퓨터전공,
(주)빅웨이브에이아이 데이터 분석팀 소속
hjjung933@gmail.com, juju020716@gmail.com, spdlvka147@gmail.com,
imscar0019@gmail.com, coolwin200@gmail.com, dongkyun@knu.ac.kr

RAGA Methodology Research using LangChain for RAG-based LLM Performance Evaluation

Alice Jung, Juhyun Song, Sanghoon Seo, Jinhyo Lim, Hyunsang Lee and Dongkyun Kim
Kyungpook National University, Bigwave AI

Abstract

The emergence of large language models (LLM) like ChatGPT is positively affecting our society, and people are hoping to adopt them in their industry. On the other hand, those are also causing problems like Hallucinations producing false information. Retrieval-augmented generation (RAG) is one of the methods that can solve hallucination problems. Therefore, we introduce the framework that can evaluate the performance of RAG methods using the RAG assessment (Ragas) evaluation framework. Using datasets generated by GPT containing question, answer, and context, LLM answers to queries. Then, Ragas evaluates the performance of each LLM and RAG method with retrieved contexts and generated answers. Ultimately, we suggest that companies planning to bring LLM into their products utilize this evaluation framework to evaluate the performance of RAG methods and LLM.

I. 서론

ChatGPT를 필두로 거대 언어 모델(Large Language Model; LLM) 관련 시장이 많은 관심을 받고 있다. 2023년 4월, 미국의 AI oberservability와 LLM evaluation 플랫폼 회사인 Arize AI가 Arize:Observe의 참가자와 머신러닝 팀을 대상으로 시행한 설문조사[1]에 따르면 53.3%의 데이터 과학자나 엔지니어들이 거대 언어 모델을 12개월 내 또는 가능한 한 빨리 자신의 생산(production)에 적용할 계획이 있다고 응답했다. 절반이 넘는 데이터 과학자나 엔지니어들이 거대 언어 모델을 사용할 의향이 있다고 밝혔으나 이들이 생각하는 생산 LLM 배포의 장벽으로 68.3%의 사람들이 데이터 프라이버시와 OpenAI로의 독점 데이터 전송 또는 내부 보안 승인의 필요성을 선택하였고, 두 번째로 43.3%의 사람들이 답변의 정확성과 환각 현상을 선택하였다.

이처럼 거대 언어 모델을 실제 생산(production)에 적용하는 것에 대해 개인 정보 보안 문제, 정확도, 환각 현상 등 여러 문제점이 제기되고 있다. 그중 하나로 환각 현상은 학습하지 않은 내용을 답변하려고 할 때, 이야기를 지어내어 답변하는 현상이다. 예를 들면, 화제가 되었던 '세종대왕 맥북 던짐 사건'은 대표적인 LLM의 환각 현상으로 인해 발생한 문제이다. 세종대왕은 맥북을 던졌던 적이 없지만 ChatGPT가 조선왕조실록에 기록된 내용처럼 답변했

[1] "본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업의 연구결과로 수행되었음" (2021-0-01082)

다[2]. 이처럼 일어나지 않았던 사건을 물어보았을 때, ChatGPT는 실제로 있었던 일인 것처럼 말을 꾸며내어 답변한다. 이러한 한계로 사용자가 잘못된 정보를 얻고, 이를 믿고 이용할 수 있다. 따라서, LLM을 생산(production)에 접목하거나 이용할 예정이 있는 회사가 사용자에게 환각 현상으로 인한 잘못된 정보를 전달하는 위험을 방지해야 할 필요가 있다.

본 논문에서는 환각 현상을 보완하기 위한 방법인 Retrieval-Augmented Generation(RAG)을 이용한 LLM의 성능을 정량적으로 평가하는 프로세스를 정의하고자 한다. 즉, LangChain을 이용하여 RAG 기법을 적용하였을 때, RAG와 LLM의 평가 프레임워크를 정립하고 수행한다. 이러한 프레임워크를 통해 RAG를 이용한 LLM의 성능 평가 과정을 확립하고 효율화한다.

II. 본론

2.1 언어모델과 LLM

자연어란 사람들이 일상적으로 사용하는 언어를 의미한다. 자연어처리는 컴퓨터 과학의 한 분야로 영어, 한국어와 같은 자연어를 컴퓨터가 이해할 수 있는 형태로 변환하는 방법론에 대해 다룬다. 이때, 자연어 처리를 수행하기 위해 만들어진 알고리즘이나 프로그램을 언어모델[3]이라고 한다. 다양한 언어모델 중에서 딥러닝 분야의 발전으로 인해 탄생한 거대 언어 모델(Large Language Model; LLM)은 방대한 자연어 데이터를 사용하여 학습된 인공지능 모델을 의미한다. 최근 부상한 LLM은 감정 분석, 고객 서비스, 챗봇, 온라인 검색 등 많은 분야에서 수요가 급증하고 있으며 그에 따른 연구가 속속 발표되고 있다. 대표적인 대규모 언어모델로는 GPT 시리즈가 있으며, 특정한 텍스트의 다음 단어를 예측하는 언어 모델을 바탕으로 만들어진 텍스트 생성기이다. GPT는 타 LLM에 비해 자연스러운 답변을 생성한다는 특징이 있다. 또 다른 대규모 언어모델인 구글의 BERT는 양방향 학습을 바탕으로 뛰어난 문장 이해 성능을 보여주는 모델이다[4][5].

본 연구는 LLM 공개 플랫폼인 허깅페이스에 공개된 언어 모델 중 실험에 사용할 그래픽 가속 장치가 수용할 수 있는 최대 80억 매개 변수를 가지고 있고, 충분한 학습으로 실험용 프롬프트와 질문에 대해 무한루프에 빠지지 않고 답변을 한 다음 세 가지 모델을 선정하였다.

Llama-3-8b-Instruct[6]는 다양한 문화 정보를 학습하고, 인류 가치에 위반되는 답변을 하지 않도록 세밀하게 조정된 Meta의 명령형 공개 LLM이다. Phi-3-mini-4k-instruct[7]는 고성능보다는 온디바이스 구동을 고려한 Microsoft의 명령형 공개 LLM이다. Mistral-7b-Instruct-v0.2[8]는 Sliding Window Attention을 통해 긴 컨텍스트를 적용할 수 있는 Mistral의 명령형 공개 LLM이다.

2.2 RAG

RAG은 왜곡될 수 있는 기억을 실제 정보로 보강하는 언어 생성 기술을 의미한다. LLM이 지식 기반 언어를 생성하기 위해 지식 기반이 되는 어떤 문서 z에 대해 LLM이 정보를 습득하는 방식은 미세조정, 학습으로 내부 파라미터 수정을 통해 파라미터에 적용된 기억 형태의 정보 습득 이외에는 없었다. RAG는 LLM이 파라미터에 종속되지 않은 문서 z 및 내부 파라미터에 적용된 기억을 융합해 언어를 생성하는 하이브리드 생성 모델이다. 이를 통해 LLM이 존재하지 않는 사실을 지어내는 듯 언어를 생성하는 현상인 환각 현상을 줄일 수 있다[9].

본 연구는 LangChain을 활용해 연구를 진행했다. LangChain이란 2022년 10월에 시작된 오픈소스 프로젝트로 LLM을 활용한 애플리케이션 개발에 쓰이는 파이썬 프레임워크이다. LangChain에서 RAG를 위해 다양한 검색기(Retriever)를 제공하는데, 벡터저장소 기반 검색기(vector store-backed retriever)의 최대 한계 관련성 검색(MMR)기법을 이용한 실험 결과와 Smaller chunks 기법으로 생성한 다중 벡터저장소 검색기(MultiVectorRetriever)를 이용해 실험한 결과를 비교해 성능평가를 진행했다.

벡터 저장소 기반 검색기(Vector Store-backed Retriever)는 데이터를 벡터 형태로 변환한 뒤 검색기에 활용하기 알맞은 벡터 저장소를 구축하고 이것을 사용해 문서를 검색하는 검색기이다. 최대 한계 관련성(Maximum Marginal Relevance; MMR) 검색 방식은 쿼리를 바탕으로 벡터 저장소 내의 데이터를 검색할 때 쿼리와 데이터 간의 유사성과 검색된 데이터 간의 다양성을 동시에 고려해 검색 결과의 품질을 향상하는 알고리즘이다. MMR은 아래 수식(2.1)을 따른다[10].

$$MMR = \lambda * Sim(d, Q) - (1-\lambda) * Sim(d, d') \quad (2.1)$$

$Sim(x, y)$: x와 y사이의 유사도

Q : 쿼리, d : 검색된 문서,

d' : 검색된 문서 중에서 d와 가장 유사한 문서,

λ : 매개변수,

$\lambda = 1$일 때 유사성만 고려,

$\lambda = 0$일 때 문서 사이의 다양성만 고려

다중 벡터저장소 검색기(MultiVector Retriever)는 하나의 데이터에서 추출된 벡터를 분할된 여러 개의 벡터저장소에 저장한 뒤 검색에 활용한다. 다중 벡터 저장소 검색기를 구축할 때 smaller chunks, summary, hypothetical Queries와 같은 구축 방법이 준비되어 있는데 이를 통해 서로 다른 특징을 가진 다중 벡터저장소를 선택해 사용할 수 있다. Smaller chunks 기법은 문서 데이터를 더 작은 단위로 분할한 뒤 각 chunk에 대해 별도의 벡터 저장소를 생성하고, 문서의 세부 정보를 더 상세히 고려한 검색 결과를 얻을 수 있는 장점이 있다[11].

2.3 Ragas

RAG Assessment(Ragas)는 외부 데이터를 활용하여 대화형 시스템에서 사용되는 RAG 파이프라인의 품질과 성능을 측정하는 프레임워크이다[12]. 이 프레임워크는 다양한 평가 지표를 제공하여 생성된 답변의 일관성, 정확성 및 의미적 일치성을 평가할 수 있다. 본 연구에서는 Faithfulness와 Answer Relevancy를 이용하여 비교하였다. 두 지표 모두 0에서 1 사이의 값을 지닌다.

Faithfulness는 제출된 답변의 충실도를 평가하여 모델의 성능을 평가한다. 답변이 실제 정답과 얼마나 충실한지를 측정한다. 생성된 답변을 개별 statement로 나누고, 각 statement에 대해 given context와 비교한다. 여기서 given context는 retrieved context로 retrieval을 통해 얻어진 context를 말한다. Faithfulness는 아래 수식(2.2)을 따른다[13].

$$\textit{Faithfulness score} = \frac{|\textit{Number of claims in the generated answer that can be inferred from given context}|}{|\textit{Total number of calims in the generated answer}|} \quad (2.2)$$

Answer Relevancy는 답변의 관련성을 평가한다. 답변이 질문과 관련이 있는지를 확인하여 점수를 부여한다. 불완전하거나 중복된 정보를 포함하는 답변은 점수가 낮게 평가된다. 생성된 답변에 대해 LLM이 몇 가지 질문을 생성한다. 이후 생성된 질문과 기존 질문에 대해 코사인 유사도를 이용하여 Answer Relevancy를 구한다. 예를 들어, Answer Relevancy는 아래 수식(2.3)을 따른다[14].

$$\textit{answer relevancy} = \frac{1}{N}\sum_{i=1}^{N}\cos(E_g, E_o) = \frac{1}{N}\sum_{i=1}^{N}\frac{E_{g_i}\cdot E_o}{\| E_{g_i} \| \| E_o \|} \quad (2.3)$$

E_g: *The embedding of the generated question i*
E_o: *The embedding of the original question*
N: *The number of generated questions* (*default*: 3)

Ⅲ. 실험

3.1. 데이터 생성 및 전처리

본 연구를 위한 데이터는 국내에 발표된 국내 OTT (Over-The-Top) 시장 관련 논문들을 이용한 질문과 답변 생성을 통해 수집되었다. 수집된 PDF 문서들은 총 6개로 국내 학술 콘텐츠 플랫폼인 'DBpia'(https://www.dbpia.co.kr)에 '한국 OTT'를 검색어로 추출한 상단 10개 문서 중 선택했다.

평가 데이터 생성을 위해 문서 내 영문 abstract을 모두 삭제하여 사용하였다. 그 이유는 영문 초록을 삭제하지 않았을 때, Question 데이터 생성 시 영어로만 질문이 생성되었기 때문이다. 한국 회사의 사업 아이템으로 이용할 수 있도록 하기 위해서 한국어로 데이터가 생성되어야 할 필요가 있기 때문에, 이를 방지하기 위하여 6개의 논문 모두 같은 과정을 거쳤다.

평가 데이터 생성은 OpenAI의 API를 이용하여 각 문서당 7~10개의 Question, Context, Answer를 생성하였다. 각 문서로부터 수백 개의 QA(Question-Context-Answer) 샘플을 수동으로 생성하는 것은 시간과 노동 집약적일 수 있다. 또한 사람이 생성한 질문은 평가에 필요한 복잡성에 도달하기 어려워 평가에 영향을 미칠 수 있다. 따라서 GPT를 이용하여 데이터를 생성하면 기존 데이터 집계 과정보다 데이터 생성 시간을 90%까지 감소시킬 수 있다. OpenAI API에서 제공하는 'gpt-3.5-turbo-16k' 모델을 통해서 질문(Question)과 문맥(Context)을 통해 답변(Answer)을 생성하고, 'gpt-4'를 활용하여 이를 한 번 더 평가(Critic)하여 샘플의 완성도를 높였다[15].

생성된 데이터의 누락 값(NaN)이 포함된 질문과 의미상으로 중복된 질문이 있는 데이터는 실험 시 적절치 않기 때문에 제거하였다. 각 문서당 5개의 서로 다른 내용의 질문을 선별하였고, 최종적으로 6개의 PDF 당 5개의 질문을 선택하여 30개의 질문으로 데이터셋을 구축하였다.

	Faithfulness	Answer Relevancy
Vector Store-backed Retriever	**0.78**	0.83
MultiVector Retriever	0.77	**0.85**

Table 3.1. RAG Methods Performance

	LLaMA3		Phi-3		Mistral	
	Faithfulness	Answer Relevancy	Faithfulness	Answer Relevancy	Faithfulness	Answer Relevancy
Vector Store-backed Retriever	0.79	**0.91**	**0.88**	0.78	0.66	0.81
MultiVector Retriever	0.75	**0.91**	0.76	0.73	**0.79**	0.90

Table 3.2. LLM Performance

최종적인 평가 데이터는 question, contexts, ground truth, evolution type으로 구성된다. question은 평가에 쓰일 질문을, context는 주어진 질문에 대한 배경이나 관련 정보를, ground truth는 생성된 답변과 비교하여 평가되는 실제 정답을, evolution type은 생성된 질문이 어떤 형식인지를 보여주며 이들은 simple, reasoning question type, multi-context으로 나누어진다. 이 외에도 metadata로 문서의 특정 정보들을 그리고 episode로 데이터의 생성이 잘 되었는지도 확인할 수 있다. 위와 같이 전처리된 데이터는 RAG 방법론과 LLM의 성능을 평가하는 데에 사용하게 된다.

3.2. 실험 방법

선정한 PDF와 3.1절에서 생성한 데이터셋을 이용하여 실험을 진행한다. 총 두 종류의 RAG 방법론(Vector Store-backed Retriever, MultiVector Retriever)과 Hugging Face에서 제공되는 세 가지의 오픈 소스 LLM (LLaMA3, Phi-3, Mistral)을 사용하여 총 여섯 가지의 실험을 진행하였다. 각 LLM은 다음과 같은 방법을 통해 answer를 출력한다. 데이터셋의 각 질문에 대해 LLM이 RAG 기법에 따라 context를 검색한다. 해당 context에 대해 LLM이 answer를 출력한다. 최종적으로 추론된 answer를 Ragas로 Faithfulness와 Answer Relevancy를 계산하여 성능을 측정한다. 이때, 평가 기준의 일관성을 위해 Faithfulness에서 사용되는 context는 retrieved context가 아닌 기존 데이터셋의 context를 기준으로 평가하였다.

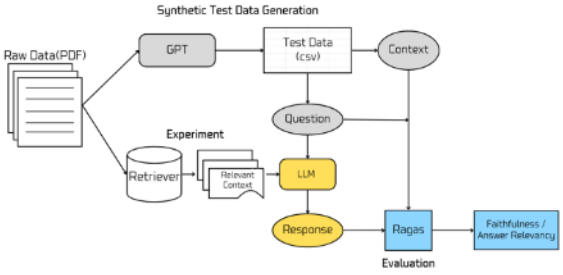

Figure 3.1. The graphical description of the process

3.3. 실험 결과

3.2절에서 제안한 실험 방법대로 세 가지 LLM과 두 가지 RAG 방법론을 사용하여 실험하였다. 본 연구에서는 RAG 방법론 간 성능을 비교하고, LLM 간 성능을 비교하였다. Ragas는 각 Question, Context, Answer에 대해 따로 성능을 측정한다. Ragas에서는 생성된 답변에 대해 개별적으로 성능을 제시한다. 따라서, 이를 비교하기 위해 RAG 방법론별 평균값과 LLM 별 평균값을 계산해 비교하였다.

3.3.1. RAG 방법론 간 성능 비교

Table 3.1은 RAG 기법 간 성능의 평균을 나타낸다. Vector Store-backed Retriever를 사용했을 때, Faithfulness 지수가 가장 높다. 이는 Vector Store-Backed Retriever가 생성한 answer를 statement로 분리하였을 때, 각 statement들이 구축된 데이터의 context와 유사함을 의미한다. 반대로, MultiVector Retriever를 사용했을 때, Answer Relevancy가 0.85로 가장 높다. MultiVector Retriever가 생성한 answer에 따라 생성된 question이 실제 question과 유사함을 의미한다. 따라서, Vector Store-backed Retriever는 context에 연관 있는 answer로 답변하지만, question에 대한

유사도가 MultiVector Retriever보다 낮다고 해석될 수 있다. 또한 MultiVector Retriever를 사용했을 때, Answer Relevancy가 높은 이유는 MultiVector Retriever가 Vector Store-backed Retriever에 비해 작은 단위로 문서를 임베딩하므로, context로부터 question에 더 근접한 의미를 찾는 데에 기여한 것으로 나타났다. 따라서 Retriever의 특징에 의해 각 성능지표 당 성능 차이가 보이는 것으로 해석된다.

3.3.2. LLM 간 성능 비교

Table 3.2는 각 오픈 소스 LLM 별 두 Retriever에 대한 성능을 나타낸다. 두 Retriever에 대한 Answer Relevancy가 모두 LLaMA3을 이용했을 때 가장 높다. 이는 곧 해당 모델이 생성한 answer를 기반으로 만들어진 question과 기존 question이 유사함을 의미한다. 그러므로 answer가 question에 연관성이 높게 생성되었음을 의미한다. 더 나아가 LLaMA3의 answer 생성 능력이 타 모델에 비해 우세하다고 해석된다. 그리고, LLaMA3의 파라미터는 80억 개로 실험에 사용된 모델 중 가장 많은 개수이므로 answer 생성 성능이 높다고 해석할 수 있다. Faithfulness의 경우, Vector Store-backed Retriever를 사용했을 때는 Phi-3에서 가장 성능이 높았고, Multivector Retriever를 사용했을 때는 Mistral에서 가장 높았다. 결론적으로 question과 관련된 answer 생성 성능이 가장 높은 모델은 Answer Relevancy가 가장 높았던 LLaMA3이다.

Ⅳ. 결론 및 향후 연구 방향

본 논문은 LLM에서 발생하는 문제점 중 한 가지인 환각 현상을 해결하는 RAG 기법에 주목했다. 증가하는 LLM 수요에 따라 이를 환각 문제 해결할 수 있는 RAG의 성능을 측정하는 프로세스의 필요성을 절감했다. 따라서, Ragas를 이용하여 효율적으로 LangChain의 RAG와 LLM의 성능을 정량적으로 평가하는 프레임워크를 제안한다. 제안한 프로세스를 적용하여 환각 현상을 감소시킬 수 있고, 결과적으로 사용자에게 잘못된 정보를 제공하는 것을 사전에 방지할 수 있다. 이는 정보를 제공하는 회사의 위험부담을 줄일 수 있고, 사용자는 올바른 정보를 얻을 수 있을 것으로 기대된다.

본 연구의 시사점은 다음과 같다. 첫 번째로 LLM 출력에서 발생하는 환각 현상 완화 정도를 비교하기 위해 LLM과 RAG 기법을 선정하여 RAG 성능을 측정하는 프레임워크를 개발했다. 두 번째로 해당 프레임워크를 실증하기 위해 직접 3가지 오픈 소스 LLM (LLaMA3, Phi-3, Mistral)과 2가지 RAG 방법론(Vector Store-backed Retriever, MultiVector Retriever)에 대해 실험을 진행하여 그 결과를 비교 분석하였다. 세 번째로, RAG 방법론 간 성능 비교에서 Faithfulness에서는 Vector Store-backed Retriever가 지수가 높게 관찰되었고, Answer Relevancy에서 MultiVector Retriever의 성능이 높게 측정되었다. 이는 각 Retriever의 차이점으로부터 기인한 것으로 해석할 수 있다. 마지막으로, 오픈소스 LLM 간 성능 비교에서는 Answer Relevancy 기준으로 두 RAG 방법론에서 모두 LLaMA3이 가장 높은 성능으로 측정되었다. 그리고 이는 LLaMA3의 Answer 생성 능력이 타 모델에 비해 우수함으로 해석된다.

본 연구는 제한적인 예산과 GPU로 인해 데이터 구축, 방법론 적용과 모델 선정에 한계가 있었다. 따라서, 본 연구에서 부족한 데이터 다양성 및 여러 가지 방법론과 모델에 따른 성능 변화를 이해하기 위해서 더 많은 데이터와 다양한 방법론 및 규모가 큰 모델에 다양한 방법론을 이용하여 실험하는 방법을 후속 연구로 진행하는 것을 제안한다. 또한, 이 프로세스를 자동화하여 RAG 성능을 평가하는 후속 연구를 기대한다.

Acknowledgement

"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학사업의 연구결과로 수행되었음" (2021-0-01082)

참고문헌

[1] Dhinakaran, Aparna. "Survey: Massive Retooling Around Large Language Models Underway." Forbes, April 26, 2023. 링크 : https://www.forbes.com/sites/aparnadhinakaran/2023/04/26/survey-massive-retooling-around-large-language-models-underway/?sh=7ef807a814a1. 액세스 날짜: 2024년 5월 10일.

[2] 한귀영. "세종대왕이 맥북을 던져?...챗GPT의 '환각'에 속지 않으려면." 한겨레, 2023. 링크 : https://n.news.naver.com/article/028/0002630035?sid=105. 액세스 날짜: 2024년 5월 10일.

[3] 박찬준, 이원성, 김윤기, 김지후, 이활석. " 초거대 언어모델 연구 동향." pp 2. 2023년.

[4] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Improving Language Understanding by Generative Pre-Training." Proceedings of the 57th Annual Meeting of the

Association for Computational Linguistics. 2018. 링크: https://www.mikecaptain.com/resources/pdf/GPT-1.pdf. 엑세스 날짜: 2024년 5월 10일.

[5] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." 2018. 링크 : https://arxiv.org/pdf/1810.04805. 엑세스 날짜 : 2024년 5월 10일

[6] "Build the future of AI with Meta Llama 3." 2024. 링크 : https://llama.meta.com/llama3/. 액세스 날짜: 2024년 5월 10일.

[7] "Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone." arXiv preprint arXiv:2404.14219, 2024. 링크 : https://arxiv.org/abs/2404.14219. 액세스 날짜: 2024년 5월 10일.

[8] "Mistral 7B." arXiv preprint arXiv:2310.06825, 2023. 링크 : https://arxiv.org/pdf/2310.06825. 액세스 날짜: 2024년 5월 10일.

[9] Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." arXiv preprint arXiv:2005.11401, 2020. 링크 : https://arxiv.org/pdf/2005.11401v4. 액세스 날짜: 2024년 5월 10일.

[10] "MMR (Maximum marginal relevance search)." 링크 : https://wikidocs.net/231585. 액세스 날짜: 2024년 5월 10일.

[11] " 다중 벡터저장소 검색기 (MultiVectorRetriever)." 링크 : https://wikidocs.net/234281. 액세스 날짜: 2024년 5월 10일.

[12] "Introduction." Ragas Documentation. 링크 : https://docs.ragas.io/en/latest/index.html. 액세스 날짜 : 2024년 5월 10일.

[13] "Faithfulness." Ragas Documentation. 링크 : https://docs.ragas.io/en/latest/concepts/metrics/faithfulness.html. 액세스 날짜: 2024년 5월 10일.

[14] "Answer Relevance." Ragas Documentation. 링크 : https://docs.ragas.io/en/latest/concepts/metrics/answer_relevance.html. 액세스 날짜: 2024년 5월 10일.

[15] "Synthetic Test Data generation." Ragas Documentation. 링크 : https://docs.ragas.io/en/stable/concepts/testset_generation.html. 액세스 날짜: 2024년 5월 10일.