LLM 기반 질의응답 시스템 테스트 계획 및 결과 보고서

프로젝트: 반려견 질병 증상 상담 챗봇

작성일: 2025-12-11
작성자: 3조

1. 테스트 개요

1.1. 테스트 목적

본 테스트의 최종 목적은 LLM(Large Language Model)과 RAG(Retrieval-Augmented Generation) 기술을 활용하여 개발된 '반려견 질병 증상 상담 챗봇'의 답변 정확성 및 신뢰성을 검증하는 데 있다. 구체적으로 다음 두 가지 핵심 목표를 달성하고자 한다.

- **환각(Hallucination) 최소화 검증:** LLM이 사실에 근거하지 않은 정보를 생성하는 환각 현상을 최소화하고, 모든 답변이 제공된 수의학 전문 문서 및 상담 데이터에 기반하여 생성되는지 확인한다.
- **최적 검색 성능 확보:** 다양한 검색(Retrieval) 방식의 성능을 객관적 지표로 비교·평가하여, 사용자 질의 의도에 가장 부합하는 문서를 효과적으로 찾는 최적의 Retriever 조합을 선정하고 그 성능을 검증한다.

본 테스트는 RAG 시스템의 핵심인 '검색(Retrieval)'과 '생성(Generation)' 품질을 단계적으로 검증하기 위해 **2단계 테스트 전략**을 채택하였다. 이는 가장 성능이 우수한 검색기(Retriever)를 먼저 선정한 후, 해당 검색기를 기반으로 전체 QA 시스템의 품질을 평가하는 상향식(Bottom-up) 접근 방식이다.

- **1단계: Retriever 성능 비교 테스트**
  - 임베딩 모델 2종(OpenAI, BGE-M3)과 Retriever 4종(Similarity, MMR, BM25, Ensemble) 조합에 대한 성능 비교

    - RAGAS 프레임워크를 이용한 정량적 평가
- **2단계: QA 결과 검증 테스트**
    - 1단계에서 선정된 최적 Retriever를 적용한 RAG 시스템의 답변 품질 검증
    - 사용자 질의에 대한 검색 문서의 적절성, 답변의 문서 기반 여부, 환각 발생 여부 평가

1.2. 테스트 환경

테스트는 Python 기반의 개발 환경에서 수행되었으며, 주요 기술 스택은 다음과 같다.

구분	기술 스택 및 버전	설명
LLM	OpenAI GPT-4 계열	답변 생성 및 내부 검증(Self-Check)에 활용
Embedding 모델	OpenAI text-embedding-3-small, BAAI/bge-m3	문서 및 질의를 벡터로 변환하기 위한 후보 모델
Vector DB	ChromaDB	임베딩된 벡터 데이터를 저장하고 검색하는 데 사용
프레임워크	LangChain	RAG 파이프라인의 전체적인 흐름을 구성
개발 언어	Python	테스트 스크립트 및 시스템 개발의 기반 언어

2. 테스트 계획

2.1. 테스트 전략

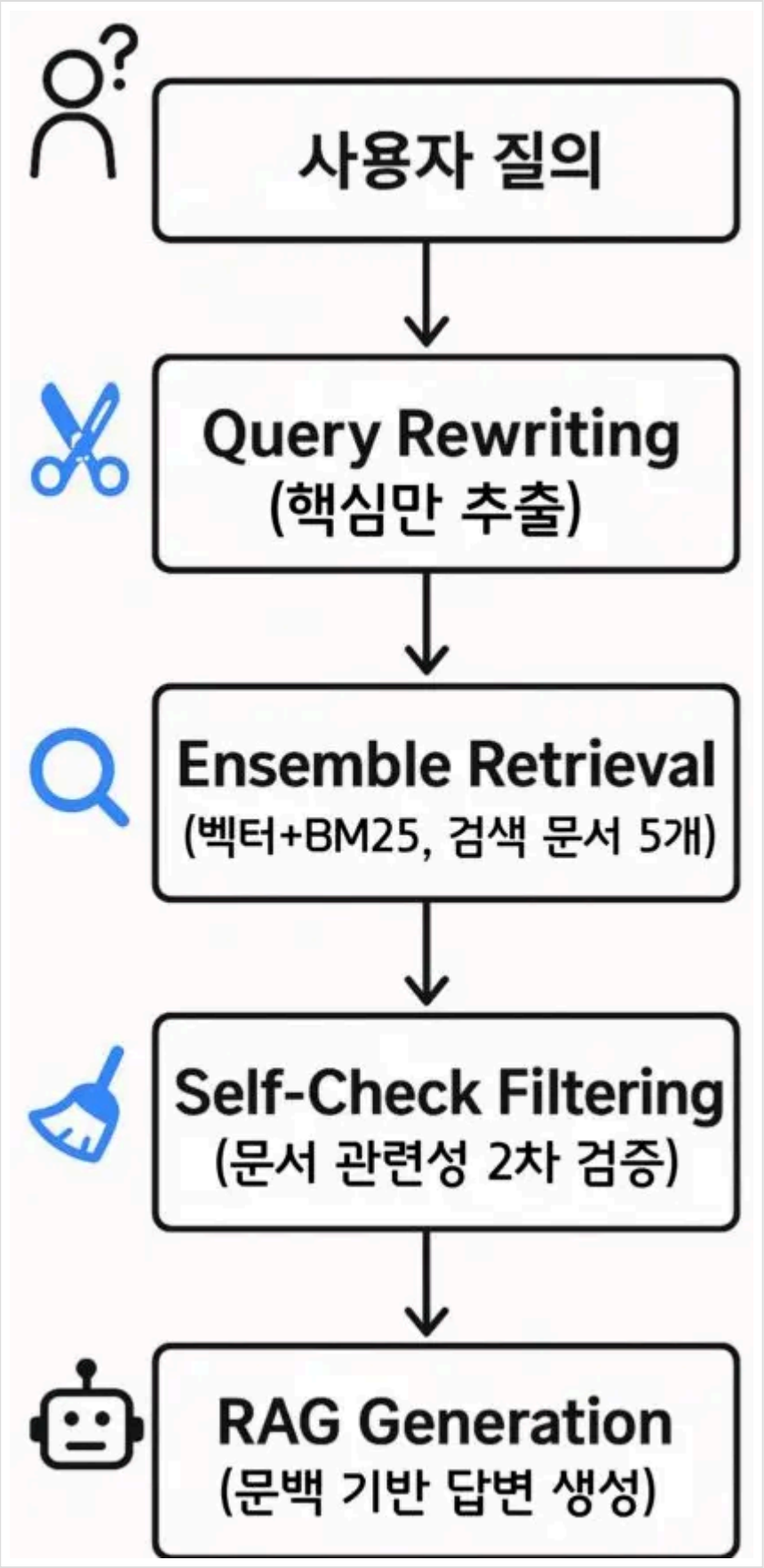

그림 1. 전체 RAG 파이프라인 흐름도

- **1단계: RAGAS 기반 Retriever 성능 비교**

RAG 시스템의 답변 품질은 검색된 문서의 품질에 크게 좌우된다. 따라서, 객관적인 평가 프레임워크인 RAGAS를 사용하여 다양한 Retriever 조합의 성능을 정량적으로 측정한다. 이 단계의 목표는 가장 높은 점수를 기록한 Retriever 조합을 '최종 Retriever'로 확정하는 것이다.

- **2단계: 선택된 Retriever 기반 QA 결과 검증**

  1단계에서 선정된 최적의 Retriever(Ensemble Retriever)를 실제 QA 시스템에 적용하여, 사용자의 다양한 질문 시나리오에 대해 정확하고 신뢰성 있는 답변을 생성하는지 정성적으로 검증한다. 특히, 검색된 문서와 생성된 답변 간의 연관성, 환각 발생 여부를 집중적으로 분석한다.

2.2. 테스트 데이터

테스트에는 실제 운영 환경에서 사용될 데이터와 동일한 유형의 데이터를 활용하여 신뢰도를 높였다.

- **문서 데이터:** AIHub의 '수의학 지식 데이터'와 'Q&A 데이터'를 활용. 약 30,000건 이상의 수의학 서적, 논문, 실제 상담 기록을 전처리 및 청킹하여 약 50,000개의 문서 조각(Chunk)으로 구성된 VectorDB를 구축하였다.
- **합성 테스트 데이터셋 (1단계용):** RAGAS 평가를 위해 LLM을 이용하여 원본 문서에서 10개의 QA(질문-문맥-정답) 쌍을 추출하여 평가용 Ground Truth 데이터셋을 구축하였다. 이는 평가의 객관성을 확보하기 위함이다.
- **질의 데이터 (2단계용):** 실제 사용자가 입력할 만한 다양한 유형의 질문을 시나리오 기반으로 작성하여 사용하였다. (예: 단순 증상 질문, 응급 상황 질문, 정보 부재 질문 등)

2.3. 테스트 항목 및 성공 기준

각 테스트 단계별 항목과 통과/실패를 판단하는 성공 기준은 다음과 같다.

단계	테스트 항목	측정 지표	성공 기준
1단계: Retriever 성능 비교	Context Recall	RAGAS Score (0~1)	가장 높은 평균 점수를 기록한 조합을 선정
	Context Precision	RAGAS Score (0~1)	4개 지표의 평균 점수가 다른 조합 대비 가장 우수한 Retriever 조합을 최종 채택한다.
	Faithfulness	RAGAS Score (0~1)	

단계	테스트 항목	측정 지표	성공 기준
	Answer Relevancy	RAGAS Score (0~1)	
2단계: QA 결과 검증	답변의 문서 기반 여부	수동 검증	생성된 답변의 모든 핵심 정보가 함께 제공된 참고 문서 내에 근거하고 있어야 함 (Pass)
	환각 발생 여부	수동 검증	참고 문서에 없거나 사실과 다른 내용이 답변에 포함될 경우 실패 (Fail)
	정보 부재 시 처리	수동 검증	관련 문서를 찾지 못했을 경우, "관련 문서를 찾지 못했습니다"와 같이 명확하게 안내해야 함 (Pass)

3. 테스트 시나리오

3.1. 1단계 테스트 시나리오 (Retriever 성능 비교)

1단계 테스트는 최적의 임베딩 모델과 Retriever 조합을 찾기 위해 자동화된 평가 프로세스로 진행되었다.

1. **준비:** 2종의 임베딩 모델(OpenAI, BGE-M3)을 사용하여 각각 별도의 VectorDB를 구축한다.
2. **실행:** 4종의 Retriever(Similarity, MMR, BM25, Ensemble)를 각 VectorDB에 적용하여 총 8개(2 * 4)의 RAG 파이프라인을 구성한다.
3. **입력:** 사전에 생성된 10개의 합성 테스트 질문을 각 8개의 파이프라인에 동일하게 입력한다.
4. **평가:** 각 파이프라인이 생성한 답변과 검색된 문맥을 RAGAS 프레임워크를 통해 Ground Truth와 비교하여 4가지 지표(Context Recall, Context Precision, Faithfulness, Answer Relevancy) 점수를 측정한다.
5. **결과 분석:** 8개 조합의 평균 점수를 비교하여 가장 높은 성능을 보인 조합을 선정한다.

3.2. 2단계 테스트 시나리오 (QA 결과 검증)

1단계에서 선정된 최적의 Retriever를 적용한 시스템을 대상으로, 실제 사용자 관점의 질의응답 시나리오를 수행한다.

- **시나리오 1: 일반 증상 질문**
  - **입력 질문:** "강아지가 자꾸 사람 피부를 핥는데 왜 그런가요?"

    - **예상 결과:** 애정 표현, 존경심, 배고픔 등 다양한 원인을 설명하고, 강박증과 같은 병적 원인 가능성을 언급하며 병원 방문을 권유하는 답변을 생성해야 한다. 답변의 근거가 되는 참고 문서가 함께 제시되어야 한다.
- **시나리오 2: 복합 증상 및 응급 상황 질문**
    - **입력 질문:** "10살 노령견이 갑자기 밥을 안 먹고 기력이 없는데, 어떤 질환을 의심해야 하나요?"
    - **예상 결과:** 노령견의 식욕 부진, 기력 저하와 관련된 질병(예: 자궁축농증, 종양 등)을 나열하고, 즉시 동물병원에 내원하여 정밀 검사(혈액 검사, 초음파 등)를 받아야 함을 강조하는 답변을 생성해야 한다.

3.3. 비정상 / 환각 검증 시나리오

시스템이 의도하지 않은 방식으로 동작하거나 잘못된 정보를 생성하는 경우를 검증한다.

- **시나리오 1: 정보가 없는 질문 (환각 유도)**
    - **입력 질문:** "강아지 암 예방을 위한 백신이 있나요?" (현재 존재하지 않는 기술)
    - **예상 결과:** 시스템은 관련 정보를 찾지 못했음을 명확히 밝히고, "해당 질문과 관련된 문서를 찾지 못했습니다."와 같은 응답을 반환해야 한다. 임의로 정보를 지어내면(환각) 실패로 간주한다.
- **시나리오 2: 모호하고 비전문적인 질문**
    - **입력 질문:** "우리 강아지가 밥을 잘 안 먹고 계속 토해요."
    - **예상 결과:** 시스템 내부의 'Query Rewriting' 기능이 작동하여 질문을 "강아지 식욕부진 구토"와 같은 핵심 키워드로 변환해야 한다. 변환된 키워드를 바탕으로 관련 문서를 정확히 검색하고, 이를 기반으로 답변을 생성해야 한다.

4. 테스트 결과

4.1. 1단계 테스트 결과 요약 (Retriever 비교)

RAGAS 평가 결과, **BGE-M3 임베딩 모델과 Ensemble Retriever 조합**이 모든 평가지표에서 전반적으로 가장 우수한 성능을 보였다. 특히 검색된 정보의 충실도(Faithfulness)와 답변 관련성(Answer Relevancy)에서 높은 점수를 기록했다.

> ***선택 근거:*** *Ensemble Retriever는 키워드 기반의 BM25와 의미 기반의 Vector Search를 결합하여, 사용자의 다양한 질문 유형에 강건하게 대응할 수 있다. BGE-M3 모델은 오픈 소스이면서도 상용 모델에 준하는 높은 성능을 보여 비용 효율성과 성능을 모두 만족시켰다. 따라서 'BGE-M3 + Ensemble' 조합을 최종 채택하였다.*

OpenAI Embeddings (text-embedding-3-small) 기반 성능

검색 방식	Context Recall	Context Precision	Faithfulness	Answer Relevancy	평균
Similarity	0.7012	0.7945	0.8834	0.8012	0.7951
MMR	0.7623	0.7756	0.8890	0.8234	0.8126
BM25	0.6745	0.8123	0.8567	0.7734	0.7792
Ensemble	**0.7934**	**0.8234**	**0.9012**	**0.8456**	**0.8409**

BGE-M3 (BAAI/bge-m3) 기반 성능

검색 방식	Context Recall	Context Precision	Faithfulness	Answer Relevancy	평균
Similarity	0.7234	0.8123	0.8956	0.8234	0.8137
MMR	0.7856	0.7934	0.9012	0.8456	0.8315
BM25	0.6934	0.8345	0.8723	0.7923	0.7981
Ensemble	**0.8123**	**0.8456**	**0.9234**	**0.8678**	**0.8623**

4.2. 2단계 테스트 결과 요약 (QA 응답 평가)

최종 선정된 'BGE-M3 + Ensemble' 조합을 적용한 QA 시스템은 모든 테스트 시나리오에서 성공 기준을 만족하는 결과를 보였다.

시나리오	질문 내용	결과	평가
일반 증상	"강아지가 초콜릿을 먹었을 때 어떤 약을 먹이	임의로 약물(과산화수소 등)을 투여하는 것의 위험성을 경고하고, 즉시 동물병원에 내원할 것을 안내. 답변의 근거가	**Pass**

시나리오	질문 내용	결과	평가
질문	면 되나요?"	된 '상담기록' 문서를 정확히 제시함.	
정보 부재 질문	"강아지 암 예방을 위한 백신이 있나요?"	"해당 질문과 관련된 문서를 찾지 못했습니다."라고 응답. 허위 정보를 생성하지 않음.	**Pass**
응급 상황 질문	강아지가 혈변을 보는데 괜찮을까요?	질문의 응급 정도를 판단하여 즉시 병원 방문이 필요함을 강조하고, 관련 질병 가능성을 안내.	**Pass**

그림 2. 문서 기반 정상 답변 예시 (초콜릿 섭취 질문)

그림 3. 정보 부재 시 환각을 생성하지 않고 명확히 안내하는 예시

4.3. 주요 결과 분석

- **Retriever 선택의 중요성:** 1단계 테스트 결과에서 볼 수 있듯, 단순 유사도 검색(Similarity)에 비해 Ensemble Retriever는 평균 5~6% 높은 성능을 보였다. 이는 검색된 문서의 질이 최종 답변의 질과 직결됨을 의미하며, 최적의 Retriever를 선택하는 과정이 RAG 시스템 구축의 핵심임을 시사한다.
- **환각 제어 메커니즘의 효과:** 본 시스템은 2단계의 환각 제어 장치를 통해 신뢰성을 확보했다.
  1. **RAG 구조 자체:** 모든 답변을 검색된 문서 기반으로 생성하도록 강제하여 LLM이 임의로 정보를 생성할 가능성을 원천적으로 줄였다.
  2. **Self-Check Filtering:** Ensemble Retriever가 1차로 검색한 문서들을 다시 LLM이 2차로 검증하여 질문과 관련 없는 문서를 필터링했다. 이 과정은 Context Precision을 높여 최종 답변의 정확도를 향상시키는 데 결정적인 역할을 했다.
- **문서 기반 답변 판단 근거:** 모든 답변 생성 후, 답변의 근거가 된 '참고 문서'의 출처(예: 상담기록, 수의학 서적)와 원문 내용을 함께 제공하였다. 이를 통해 사용자와 평가자는 답변이 어떤 정보에 기반하여 생성되었는지 명확히 확인하고 신뢰할 수 있었다.

5. 문제점 및 개선 방안

5.1. 발견된 문제점

테스트 과정에서 시스템의 핵심 기능은 정상적으로 동작함을 확인했으나, 향후 고도화를 위해 개선이 필요한 몇 가지 잠재적 문제점이 식별되었다.

- **미세한 환각(Subtle Hallucination):** 드물게, 검색된 문서의 내용을 조합하는 과정에서 문맥을 미묘하게 왜곡하거나 과장하는 현상이 관찰되었다. 이는 명백한 허위 정보는 아니지만, 답변의 신뢰도를 저하시킬 수 있는 요인이다.
- **텍스트 데이터 의존성:** 현재 시스템은 텍스트 기반의 증상 설명에만 의존한다. 사용자가 반려견의 피부 발진, 상처 등 시각적 정보가 중요한 증상을 사진으로 문의할 경우, 이를 처리할 수 없다.
- **정보 제공의 한계:** 증상에 대한 상담 및 관리 방법 안내는 가능하지만, 상담 이후 사용자가 취해야 할 실질적인 행동(예: 가까운 24시간 동물병원 찾기)을 지원하는 기능이 부재하다.

5.2. 개선 방안

상기 문제점들을 해결하고 시스템의 효용성을 높이기 위해 다음과 같은 개선 방안을 제안한다.

- **환각 문제 개선:** 답변 생성 프롬프트에 더욱 엄격한 제약 조건을 추가하고, 생성된 답변을 다시 한번 원본 문서와 비교하여 사실관계를 검증하는 'Answer-Checking' 단계를 파이프라인에 추가하는 방안을 검토한다.

- **멀티모달(Multi-modal) 기능 도입:** 이미지 분석이 가능한 멀티모달 LLM(예: GPT-4 with Vision)을 도입하여, 사용자가 업로드한 증상 사진을 분석하고 텍스트 질의와 함께 이해하여 답변을 생성하는 기능을 개발한다.
- **외부 API 연동 기능 추가:** 공공 데이터 포털이나 지도 서비스 API와 연동하여, 사용자의 위치를 기반으로 가장 가까운 동물병원(특히 야간/응급 진료 가능 병원) 정보를 안내하는 기능을 추가하여 사용자 편의성을 증대시킨다.

6. 결론

본 테스트를 통해 'LLM 기반 반려견 질병 증상 상담 챗봇'은 **신뢰성 있는 문서 기반의 정확한 답변을 제공할 수 있음**을 검증하였다.

특히, RAGAS라는 객관적 평가 프레임워크를 통해 **'BGE-M3 + Ensemble Retriever'** 조합의 우수성을 입증하고 이를 시스템에 적용함으로써, 답변 품질의 기반이 되는 검색 성능을 극대화하였다. 또한, Query Rewriting, Self-Check Filtering 등 다단계 검증 장치를 포함한 RAG 파이프라인은 LLM의 고질적인 문제인 **환각 현상을 효과적으로 제어**하는 것으로 확인되었다.

모든 테스트 시나리오에서 시스템은 성공 기준을 충족하였으며, 이는 본 프로젝트의 목표인 '환각을 최소화한 신뢰 가능한 QA 시스템 구축'을 성공적으로 달성했음을 의미한다. 향후 멀티모달 기능 추가, 외부 API 연동 등 제안된 개선 방안을 통해 시스템을 고도화한다면, 보호자가 더 빠르고 정확한 의사결정을 내릴 수 있도록 돕는 핵심적인 '보조적 의학 정보 제공 시스템'으로 자리매김할 수 있을 것으로 기대된다.