

# RAGRAS를 사용한 평가



# 1. RAGRAS 개요



**RAGAS**는 **RAG(Retrieval-Augmented Generation)** 시스템의 응답 품질을 자동으로 평가하기 위한 프레임워크이다.

즉, “RAG 모델이 얼마나 정확하고, 관련성 있고, 신뢰할 만한 답을 내는가?”를 정량적으로 평가한다.

## 1. 개념 요약

항목	설명
이름	RAGAS (Retrieval-Augmented Generation Assessment Suite)
목적	RAG 시스템의 성능을 정량적 메트릭으로 평가
기반 라이브러리	<a href="#">Hugging Face Evaluate</a> ↗, [Datasets]
개발 주체	OpenAI 및 HuggingFace 커뮤니티 협업 프로젝트
주요 사용처	RAG 파이프라인 품질 비교, 문서 검색 정확도 평가, 답변 신뢰도 분석

## 2. RAGAS가 평가하는 주요 메트릭

메트릭 이름	설명	측정 대상
Faithfulness (신뢰도)	답변이 실제 검색된 문맥(Context)과 일치하는지 평가	LLM이 문맥을 왜곡하지 않았는가
Answer Relevance (답변 관련성)	답변이 사용자의 질문과 얼마나 관련 있는가	답변-질문 간 의미적 일치도
Context Relevance (문맥 관련성)	검색된 문서가 질문과 얼마나 관련 있는가	검색 단계의 품질
Context Precision (문맥 정밀도)	불필요한 문서를 얼마나 포함했는가	검색 결과의 정확도
Answer Correctness (정답률, 일부 버전)	정답 텍스트와의 의미적 유사도	ground_truth 대비 품질

### 3. 평가 입력 구조

RAGAS는 `datasets.Dataset` 객체를 입력으로 사용한다.

데이터셋에는 다음 필드가 포함되어야 한다.

필드 이름	설명
<code>question</code>	사용자의 질문
<code>contexts</code>	RAG이 참조한 문서 목록
<code>answer</code>	모델이 생성한 답변
<code>ground_truth</code>	올바른 정답 또는 기준 답변

예시:

python

```
from datasets import Dataset

data = {
    "question": ["딥러닝이란 무엇인가?"],
    "contexts": [["딥러닝은 인공지능을 사용하는 머신러닝의 한 분야이다."]],
    "answer": ["딥러닝은 머신러닝의 하위 분야로 신경망을 사용한다."],
    "ground_truth": ["딥러닝은 인공지능을 기반으로 한 머신러닝 기술이다."]
}

dataset = Dataset.from_dict(data)
```

## 4. 평가 수행 예시

python

📄 코드 복사

```
from ragas import evaluate
from ragas.metrics import faithfulness, answer_relevance, context_precision, context_relevance

metrics = [faithfulness, answer_relevance, context_precision, context_relevance]

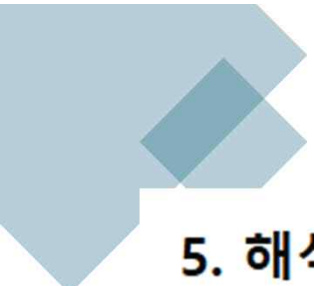
results = evaluate(dataset=dataset, metrics=metrics)
print(results)
```

출력 예시:

bash

📄 코드 복사

```
{'faithfulness': 0.93,
 'answer_relevance': 0.91,
 'context_relevance': 0.98,
 'context_precision': 0.87}
```



## 5. 해석 방법

점수 범위

해석

0.9 이상

매우 우수 — RAG이 사실적으로 정확하고 일관된 답변

0.7~0.9


양호 — 일부 불필요 문맥이나 표현 오류 존재

0.5~0.7

보통 — 검색 품질 또는 답변 신뢰도 개선 필요

0.5 미만

부족 — 문맥 불일치, 질문과 무관한 답변 등 심각한 오류 가능





## 6. LangChain과의 통합

- LangChain의 `create_agent()` 로 만든 RAG 파이프라인 결과를 RAGAS에 전달하여 평가 가능
- `LangSmith` 를 연동하면 실험별 RAGAS 자동 평가 로그 생성 가능
- `ragas.evaluate()` 는 벡터스토어/프롬프트/LLM 조합 성능 비교에 유용

## 7. 간단 요약

항목	요약
정의	RAG 시스템의 품질을 자동 평가하는 라이브러리
주요 메트릭	Faithfulness, Answer Relevance, Context Relevance, Context Precision
활용 목적	검색-생성 파이프라인의 품질 진단 및 개선
통합 가능	LangChain, LangGraph, LangSmith 등

## 2. RAG + RAGAS 평가 자동화 에이전트 구현

---

## 1 환경 및 모델 설정

python

```
llm = ChatOpenAI(model="gpt-4o-mini", temperature=0)
embedding = OpenAIEmbeddings(model="text-embedding-3-small")
```

- **ChatOpenAI** : 질문에 대한 답변을 생성하는 LLM
- **OpenAIEmbeddings** : 문서를 벡터로 변환해 검색 가능하게 만들
- **temperature=0** : 일관된(랜덤 없음) 결과 생성

## 2 문서 준비 및 벡터 DB 구성

python

```
docs = [Document(page_content="..."), ...]
db = Chroma.from_documents(docs, embedding)
```

- **문서(Document)** 5개를 준비 → 각각 AI 개념(인공지능, 머신러닝, 딥러닝 등)을 설명
- **Chroma DB** : LangChain의 벡터 저장소 → 임베딩 기반 검색 지원

### 3 검색 Tool 정의

python

```
@tool
def semantic_retrieve(query: str) -> str:
    retriever = db.as_retriever(search_kwargs={"k": 3})
    results = retriever.invoke(query)
    return "\n".join([r.page_content for r in results])
```

- `@tool` : LangChain 에이전트가 쓸 수 있는 함수로 등록
- 벡터스토어에서 유사 문서 3개(k=3) 검색
- 검색된 문서를 한 문자열로 반환

### 4 에이전트 생성

python

```
rag_agent = create_agent(model=llm, tools=[semantic_retrieve], system_prompt="...")
```

- LLM + 검색 Tool 을 묶은 RAG 에이전트
- **System prompt** 에서 "문서를 근거로만 답변하고 추측하지 말라" 규칙 설정

## 5 질의 실행 및 결과 수집

python

```
for q in questions:
    result = rag_agent.invoke({"messages": [{"role": "user", "content": q}]})
```

- 총 5개 질문을 순차 입력
- 각 질문마다 검색→답변 생성 → 콘솔 출력
- `answers[]`, `contexts[]` 리스트에 저장

## 6 RAGAS 평가 데이터 준비

python

```
data = {"question": ..., "contexts": ..., "answer": ..., "ground_truth": ...}
dataset = Dataset.from_dict(data)
```

- 각 질문에 대해
    - 질문(question)
    - 검색된 문맥(contexts)
    - 에이전트 답변(answer)
    - 정답(ground\_truth)
- 를 묶어 `datasets.Dataset` 형태로 준비

## 7 RAGAS 평가

python

코

```
metrics = [Faithfulness(), ResponseRelevancy(), ContextPrecision(), ContextRecall()]
results = evaluate(dataset=dataset, metrics=metrics)
```

### 평가 지표 의미

메트릭	의미	해석
<b>Faithfulness</b>	답변이 문서 내용에 얼마나 사실적으로 부합하는가	거짓 없을수록 높음
<b>ResponseRelevancy</b>	질문과 답변 간 의미적 연관성	핵심 질문에 얼마나 집중했는가
<b>ContextPrecision</b>	검색된 문서 중 정말 유용한 문서의 비율	불필요 문맥 적을수록 높음
<b>ContextRecall</b>	정답에 필요한 문서를 얼마나 잘 검색했는가	필요 문맥 누락 없을수록 높음

## 8 평가 출력 및 요약

python

```
print(results)
df_results = pd.DataFrame([results])
```

- `results` → 전체 평균 점수 (예: `faithfulness` 0.65등)
- `df_results` → 판다스로 변환해 정리용 표 생성
- 질문별 세부 점수는 `results.to_pandas()` 또는 `dir(results)` 로 추가 조회 가능

=== RAGAS 평가 결과 ===

```
{'faithfulness': 0.6500, 'answer_relevancy': 0.6527, 'context_precision': 0.2233, 'context_recall': 0.8000}
```

## 9 실행 흐름 요약

SCSS

문서 → 벡터스토어 (Chroma)



검색 Tool (semantic\_retrieve)



LangChain Agent (create\_agent)



질문별 답변 생성



RAGAS 평가 (Faithfulness 등)



평가 결과 DataFrame 출력




```

1 # 질문별 상세 평가 결과
2 # 평가 결과를 DataFrame으로 변환
3 df = results.to_pandas()
4 df

```

	user_input	retrieved_contexts	response	reference	faithfulness	answer_relevancy	context_precision	context_recall
0	딥러닝이란 무엇인가?	[인공지능은 인간의 지능을 모방하는 기술이며, 다양한 분야에서 활용된다., 머신러닝...	딥러닝은 인공지능망 기반의 머신러닝 기법으로, 주로 이미지 및 음성 인식 분야에서 ...	딥러닝은 인공지능망을 사용하는 머신러닝의 하위 분야이다.	0.75	0.840061	0.333333	1.0
1	자연어처리는 어떤 기술인가?	[인공지능은 인간의 지능을 모방하는 기술이며, 다양한 분야에서 활용된다., 머신러닝...	자연어처리(Natural Language Processing, NLP)는 인간의 언...	자연어처리는 인간의 언어를 이해하고 생성하는 인공지능 기술이다.	0.75	0.806033	0.250000	1.0
2	강화학습은 어떤 원리로 동작하는가?	[인공지능은 인간의 지능을 모방하는 기술이며, 다양한 분야에서 활용된다., 머신러닝...	강화학습은 보상 기반으로 학습하여 행동을 최적화하는 AI 분야입니다. 에이전트는 환...	강화학습은 보상 기반으로 학습하여 최적의 행동을 선택하는 방법이다.	0.25	0.773561	0.200000	1.0
3	머신러닝과 딥러닝의 차이점을 설명해주세요.	[인공지능은 인간의 지능을 모방하는 기술이며, 다양한 분야에서 활용된다., 머신러닝...	머신러닝과 딥러닝의 차이점은 다음과 같습니다:WnWn1. **정의**:Wn - ...	딥러닝은 머신러닝의 일부로, 신경망을 사용한다.	0.50	0.844088	0.333333	1.0
4	AI는 인간의 사고를 완전히 대체할 수 있는가?	[인공지능은 인간의 지능을 모방하는 기술이며, 다양한 분야에서 활용된다., 머신러닝...	현재의 인공지능(AI)은 인간의 사고를 모방하고 특정 작업을 수행하는 데 매우 유용...	AI는 인간 사고를 완전히 대체하기 어렵다.	1.00	0.000000	0.000000	0.0



## 분석 결과 의미

이 코드는 “RAG 시스템의 성능을 정량적으로 평가하는 자동화 파이프라인”이다.  
한 번 돌리면 각 질문 대해 검색 → 답변 → 평가 → 점수 요약 이 모두 자동 처리된다.

### 항목

### 의미

강점


RAG + 평가 통합 워크플로우, LangChain 최신 버전 호환

약점

질문별 세부 점수 출력 로직이 단순, RAGAS 평균만 표시

개선

`results.to_pandas()` 활용해 질문별 세부 점수 표 및 시각화 추가





감사합니다