

섹션 6: LLM 답변 생성 및 평가

- 패키지 설치
 - conda

```
pip install langchain_anthropic langchain_google_genai langchain_groq
```

- poetry

```
poetry add langchain_anthropic langchain_google_genai langchain_groq
```

1. LLM 모델 공급자

1.1 OpenAI GPT API

- OpenAI의 GPT 모델은 현재 가장 널리 알려진 LLM 중 하나
- 주요 특징:
 - 다양한 모델 제공: GPT-3.5, GPT-4 등 다양한 버전 제공
 - 높은 성능: 텍스트 생성, 번역, 요약 등 다양한 태스크에서 우수한 성능
 - 사용 편의성: 잘 문서화된 API와 다양한 라이브러리 지원

1.2 Anthropic Claude API

- Anthropic의 Claude는 윤리적이고 안전한 AI 개발을 목표로 하는 모델
- 주요 특징:
 - 윤리적 고려: 편향성 감소와 안전성 강화에 중점
 - 긴 컨텍스트 처리: 대량의 텍스트 입력 처리 가능
 - 다중 턴 대화: 대화 맥락을 잘 유지하며 일관성 있는 응답 생성

1.3 Google Gemini API

- Google의 Gemini는 멀티 모달 기능을 갖춘 최신 LLM
- 주요 특징:
 - 다중 모달 처리: 텍스트, 이미지, 오디오 등 다양한 입력 처리 가능
 - 효율성: 빠른 응답 속도와 자원 효율적 설계
 - 멀티태스킹: 다양한 AI 태스크를 단일 모델로 처리 가능




1.4 Ollama - 오픈소스 LLM

- Ollama는 로컬 환경에서 다양한 오픈소스 LLM을 쉽게 실행할 수 있게 해주는 프레임워크
- 주요 특징:
 - 로컬 실행: 클라우드 API에 의존하지 않고 로컬에서 모델 실행 가능
 - 다양한 모델 지원: Llama 2, Mistral, Vicuna 등 다양한 오픈소스 모델 지원
 - 커스터마이징: 모델 미세 조정 및 사용자 정의 가능

1.5 Groq API - 초고속 LLM 추론

- Groq는 고성능 AI 칩을 사용하여 초고속 LLM 추론을 제공하는 서비스
- 주요 특징:
 - 초고속 처리: 매우 빠른 응답 시간 제공
 - API 호환성: OpenAI API와 유사한 인터페이스로 쉬운 통합
 - 다양한 모델 지원: LLaMA 2, Mixtral 등 다양한 모델 제공

2. RAG 답변 평가 - Heuristic Evaluation

 Embedding Distance 텍스트를 벡터 공간에 매핑하고 거리를 측정하여 유사도를 평가 장점: <ul style="list-style-type: none">계산 효율성이 높음대규모 데이터셋에 적합 단점: <ul style="list-style-type: none">문맥 이해에 한계가 있음미세한 의미 차이 포착이 어려움	 Cross-Encoder 두 텍스트를 동시에 입력받아 직접적인 관련성 점수를 출력 장점: <ul style="list-style-type: none">높은 정확도문맥을 더 잘 이해함 단점: <ul style="list-style-type: none">계산 비용이 높음대규모 데이터셋에 적용하기 어려움	 Rouge Metric 생성된 텍스트와 참조 텍스트 간의 겹치는 n-gram을 기반으로 평가 장점: <ul style="list-style-type: none">텍스트 생성 작업에 널리 사용됨이해하기 쉽고 해석이 직관적 단점: <ul style="list-style-type: none">단어 순서나 동의어를 고려하지 않음문맥적 의미를 완전히 캡처하지 못함
---	--	--

2.1 Embedding Distance

- 임베딩 거리는 두 텍스트의 의미적 유사성을 벡터 공간에서 측정하는 방법
- 작동 원리:
 - 예측 문자열과 참조 레이블 문자열을 벡터로 임베딩
 - 두 벡터 간의 거리(보통 코사인 유사도나 유클리디안 거리)를 계산
 - 거리 점수가 낮을수록 의미적으로 더 유사함을 의미
- 장점:
 - 의미적 유사성 캡처: 단순 문자열 비교보다 더 깊은 의미 관계를 포착
 - 효율성: 한 번 임베딩을 생성하면 빠르게 비교
- 단점:
 - 컨텍스트 제한: 전체 문맥을 완전히 캡처하지 못할 수 있음

2.2 Cross-Encoder 활용

- Cross-Encoder는 두 문장을 동시에 처리하여 직접적인 유사성 점수를 제공
- 작동 원리:
 - 두 문장을 하나의 입력으로 모델에 제공
 - 모델이 두 문장의 관계를 직접 분석하여 유사성 점수를 출력
- 장점:
 - 높은 정확도: 두 문장의 관계를 직접 모델링하여 더 정확한 유사성 평가가 가능
 - 컨텍스트 인식: 전체 문맥을 고려한 평가가 가능
- 단점:
 - 계산 비용: 각 문장 쌍마다 별도의 모델 실행이 필요하여 처리 시간이 긴 편

2.3 ROUGE Metric

- ROUGE[Recall-Oriented Understudy for Gisting Evaluation]
- 텍스트 요약의 품질을 평가하는 데 널리 사용되는 메트릭
- 작동 원리:
 - ROUGE-N: N-gram 중첩을 측정
 - ROUGE-L: 최장 공통 부분 수열(LCS)을 기반으로 유사성을 측정
 - ROUGE-W: 가중치가 부여된 LCS를 사용
- 장점:
 - 빠른 계산: 단순한 단어 중첩 기반으로 빠르게 계산할 수 있음
 - 해석 용이성: 결과를 쉽게 이해하고 해석 가능
 - 표준화: 텍스트 요약 분야에서 널리 사용되는 표준 메트릭
- 단점:
 - 표면적 유사성: 깊은 의미적 관계를 포착하는 데 한계가 있음
 - 어순 민감성: 단어 순서 변경에 민감할 수 있음

3. RAG 답변 평가 - LLM as judge

🧠 LLM-as-judge 개념

LLM을 활용하여 인간과 유사한 판단을 제공하는 평가 방식

특징:

- 깊은 의미적 관련성 포착 가능
- 맥락을 고려한 정교한 평가
- 유연한 평가 기준 적용 가능

📊 전통적 메트릭과의 비교

ROUGE	LLM-as-judge
• 단어 중첩 기반	• 의미 기반 평가
• 빠른 계산	• 상대적으로 느림
• 표면적 유사성	• 깊은 의미적 유사성

✅ 장점

- 깊은 의미적 관련성 포착
- 맥락을 고려한 평가 가능
- 복잡한 평가 기준 적용 가능
- 인간의 판단과 유사한 결과

❌ 단점

- 완전한 객관성 보장 어려움
- API 사용에 따른 비용 발생
- 처리 시간이 상대적으로 길어짐
- LLM의 편향이 결과에 영향 줄 수 있음

2.1 LLM-as-judge 개요

- LLM을 평가자로 사용하는 방식은 인간의 판단에 가까운 평가를 제공
- 단순한 메트릭으로는 포착하기 어려운 의미적 관련성, 맥락 이해, 논리적 일관성 등 평가

- 장점:
 - 깊은 의미 이해: 표면적인 단어 일치를 넘어선 평가 가능
 - 맥락 고려: 질문의 의도와 답변의 적절성을 종합적으로 판단
 - 유연성: 다양한 평가 기준을 적용할 수 있음
- 단점:
 - 주관성: LLM의 판단이 완전히 객관적이지 않을 수 있음
 - 비용: API 호출에 따른 비용 발생
 - 시간: 처리 시간이 더 오래 걸릴 수 있음

2.2 LangChain - QA Evaluation

- 사용자 질문에 대한 정확성, 관련성을 평가 (Y/N, 0/1)
- 유형
 1. "qa" 평가기:
 - 목적: 사용자 질문에 대한 응답의 정확성을 직접적으로 평가
 - 방법: LLM에게 참조 답변을 기반으로 응답을 "정확함" 또는 "부정확함"으로 평가하도록 지시
 - 사용 사례: 명확한 정답이 있는 간단한 질문-답변 쌍에 적합
 2. "context_qa" 평가기:
 - 목적: 더 넓은 맥락을 고려하여 응답의 정확성을 평가
 - 방법: LLM 체인에게 예제 출력을 통해 제공된 참조 "컨텍스트"를 사용하여 정확성을 판단하도록 지시
 - 사용 사례: 정확한 정답은 없지만 관련 문서나 정보가 있는 경우에 유용
 3. "cot_qa" 평가기:
 - 목적: 체인 오브 쏫트(Chain of Thought) 추론을 통해 더 심층적인 평가 수행
 - 방법: "context_qa" 평가기와 유사하지만, 최종 판단을 내리기 전에 단계별 추론 과정을 거치도록 LLM 체인에 지시
 - 사용 사례: 복잡한 질문이나 더 깊은 분석이 필요한 경우에 적합

2.3 LangChain - Criteria Evaluation (No lables)

- 참조 레이블(ground truth)이 없는 상황에서 모델 출력의 품질을 평가
- 유형:
 1. "criteria" 평가기:
 - 목적: 주어진 기준에 따라 예측이 기준을 만족하는지 평가

- 출력: 이진 점수 (예: Yes/No 또는 1/0)
- 사용 사례: 특정 조건 충족 여부를 확인할 때 유용

2. "score_string" 평가기:

- 목적: 주어진 기준에 따라 예측의 품질을 수치로 평가
- 출력: 수치 점수 (기본적으로 1-10 척도)
- 사용 사례: 출력의 품질을 더 세밀하게 평가하고 싶을 때 유용

• 기본 "Criteria":

- 'conciseness', 'relevance', 'correctness', 'coherence', 'harmfulness', 'maliciousness',
- 'helpfulness', 'controversiality', 'misogyny', 'criminality', 'insensitivity'

2.4 LangChain - Criteria Evaluation (With lables)

- 참조 레이블(ground truth)이 주어진 상황에서 모델 출력의 품질을 평가

• 유형:

1. "labeled_criteria" 평가기:

- 목적: 참조 레이블을 고려하여 예측이 주어진 기준을 만족하는지 평가
- 출력: 이진 점수 (예: Yes/No 또는 1/0)
- 사용 사례: 참조 답변과 비교하여 특정 조건 충족 여부를 확인할 때 유용

2. "labeled_score_string" 평가기:

- 목적: 참조 레이블과 비교하여 예측의 품질을 수치로 평가
- 출력: 수치 점수 (기본적으로 1-10 척도)
- 사용 사례: 참조 답변과 비교하여 출력의 품질을 더 세밀하게 평가하고 싶을 때