
- 출처: LangChain 공식 문서 또는 해당 교재명
- 원본 URL: https://smith.langchain.com/hub/teddynote/summary-stuff-documents

---

3. `생성한 평가용 데이터셋 업로드` *(HuggingFace Dataset)*

- **`평가 데이터 세트`**
  - `AI` 애플리케이션 성능, 기능을 평가가기 위해 설계된 동질적인 `데이터 샘플` 모음
    - *data-sample: 특정 시나리오에서 `LLM` 애플리케이션의 성능을 평가하고 측정하는 데 사용되는 단일 구조화된 데이터 인스턴스*
      - *AI 애플리케이션이 처리할 것으로 예상되는 단일 상호작용 단위 or 특정 사용 사례*
      - *`RAGAS` 의 평가샘플: `SingleTurnSample`, `MultiTurnSample` 클래스로 사용하여 표현됨*
  - 구조 (구성)
    - **`샘플`**
      - `SingleTurnSample` or `MultiTurnSample` 인스턴스의 모음
      - 각 샘플 = 고유한 상호작용 or 시나리오
    - **`일관성`**
      - 평가의 일관성을 유지하기 위해 데이터 세트 내의 모든 샘플 = **`동일한 유형`** *(`모두 단일 턴 샘플` or `모두 다중 턴 샘플`)* 이어야 함
  - 효과적인 평가 데이터세트 큐레이션을 위한 지침
    - **`명확한 목표 정의`**
      - 평가하려는 AI 애플리케이션의 `구체적인 측면`과 `테스트`하려는 `시나리오`를 `파악`하기
      - 이러한 `목표`를 `반영`하는 `데이터 샘플`을 `수집`하기
    - **`대표 데이터 수집`**
      - `AI 애플리케이션`에 대한 `포괄적인 평가`를 제공하기 위해 데이터 세트가 `다양한 시나리오`, `사용자 입력` 및 `예상 응답`을 `포괄`하는지 `확인`하기
      - `다양한 소스`에서 `데이터`를 `수집`하거나 `합성 데이터`를 `생성`하여 달성 가능
    - **`품질 및 크기`**
      - 목표: `유의미한 통찰력`을 제공할 만큼 충분히 `크지만`, `다루기 힘들 정도로 크지 않은 데이터 세트`
      - 데이터가 `고품질`이고 평가하려는 `실제 상황`을 `정확하게 반영`하는지 확인

- **`Ragas 구성 요소`**
  - **`프롬프트 객체`**: 다양한 지표 및 합성 데이터 생성 작업 내에서 사용
    - **`Instruction`** (`지시`)
      - 모든 프롬프트의 기본 요소인 지시는 언어 모델(LLM)이 수행해야 하는 작업을 명확하게 설명하는 자연어 지시어
      - instruction프롬프트 객체 내의 변수를 사용 → 지정
    - **`Few-Shot Examples`** (`소수 예시`)
      - `LLM`은 소수 예시를 제공받을 때 더 나은 성능을 보이는 것으로 알려져 있음
      - `Few-shot Examples`: 모델이 작업 맥락을 이해하고 더 정확한 응답을 생성하는 데 도움이 되기 때문
        - **`examples프롬프트 객체의 변수`를 사용 → 지정**
        - 각 예시는 `입력`과 `해당 출력`으로 구성되며, `LLM`은 이를 사용하여 작업을 `학습`함
    - **`Input Model`** (`입력 모델`)
      - 모든 프롬프트는 `입력`을 통해 `출력`을 `생성`
      - Ragas에서 이 입력의 예상 형식: **`input_model변수`를 사용 → 정의**
      - 입력의 구조를 개략적으로 보여주는 `Pydantic 모델`: 프롬프트에 제공된 `데이터의 유효성 검사` 및 `구문 분석 가능`
    - **`Output Model`** (`출력 모델`)
      - 프롬프트 실행 → 출력 생성됨
      - 이 출력의 형식: **`output_model프롬프트` 객체의 변수 사용 → 지정**

        - 입력 모델과 마찬가지로 출력 모델은 `출력의 구조`를 정의하는 `Pydantic 모델` : `LLM`에서 생성된 데이터의 `유효성 검사` 및 `구문 분석`을 용이하게 함

*

- **`평가 샘플`**
    - **`SingleTurnSample`**
        - 사용자, LLM, 그리고 예상 평가 결과 간의 `단일 턴 상호작용`을 나타냄
        - 단일 질문과 답변 쌍을 포함하는 평가에 적합하며, 추가적인 맥락이나 참조 정보가 포함됨
        - 코드 예시

```
from ragas import SingleTurnSample

# User's question
user_input = "What is the capital of France?"

# Retrieved contexts (e.g., from a knowledge base or search engine)
retrieved_contexts = ["Paris is the capital and most populous city of France."]

# AI's response
response = "The capital of France is Paris."

# Reference answer (ground truth)
reference = "Paris"

# Evaluation rubric
rubric = {
    "accuracy": "Correct",
    "completeness": "High",
    "fluency": "Excellent"
}

# Create the SingleTurnSample instance
sample = SingleTurnSample(
    user_input=user_input,
    retrieved_contexts=retrieved_contexts,
    response=response,
    reference=reference,
    rubric=rubric
)
```

    - **`MultiTurnSample`**
        - `인간`, `AI`, 그리고 `선택적으로 도구 간`의 `다중 턴 상호작용`과 `평가`를 위한 예상 결과
        - 평가를 위해 더 복잡한 상호작용에서 대화형 에이전트를 표현하는 데 적합함
            - **`user_input` 속성** = `인간 사용자`와 `AI` 시스템 간의 다중 턴 대화를 구성하는 *`일련의 메시지`*
            - *`일련의 메시지`* = **`HumanMessage`**, **`AIMessage`**, **`ToolMessage`** **class**
        - 코드 예시

```
# 뉴욕시의 현재 날씨를 알고 싶어하는 사용자 → AI 비서 → 날씨 API 도구 사용 → 정보 가져옴 → 사용자에게 응답하기

from ragas.messages import HumanMessage, AIMessage, ToolMessage, ToolCall

# User asks about the weather in New York City
user_message = HumanMessage(content="What's the weather like in New York City today?")

# AI decides to use a weather API tool to fetch the information
ai_initial_response = AIMessage(
    content="Let me check the current weather in New York City for you.",
    tool_calls=[ToolCall(name="WeatherAPI", args={"location": "New York City"})]
)

# Tool provides the weather information
tool_response = ToolMessage(content="It's sunny with a temperature of 75°F in New York City.")

# AI delivers the final response to the user
```

```
  ai_final_response = AIMessage(content="It's sunny and 75 degrees Fahrenheit in New York City today.")

  # Combine all messages into a list to represent the conversation
  conversation = [
      user_message,
      ai_initial_response,
      tool_response,
      ai_final_response
    ]

  # 대화를 사용하여 참조 응답과 평가 기준을 포함한 MultiTurnSample 객체 만들기

  from ragas import MultiTurnSample

  # Reference response for evaluation purposes
  reference_response = "Provide the current weather in New York City to the user."

  # Create the MultiTurnSample instance
  sample = MultiTurnSample(
      user_input=conversation,
      reference=reference_response,
  )
```

*

- **`평가 데이터 세트`**
  - `AI 애플리케이션`의 성능과 기능을 평가하기 위해 설계된 `동질적인 데이터 샘플 모음`
  - `Ragas` 평가 데이터 세트: `EvaluationDataset` 클래스 사용 → 표현 → 평가 목적으로 데이터 샘플을 구성하고 관리하는 체계적인 방법 제공
  - **`SingleTurnSample`** → **`평가 데이터 세트 만들기`**: *개별 샘플 생성 → 데이터세트로 조립 → 기본 작업 수행*
    - **a.** **`필요한 클래스 가져오기`**: 모듈에서 `SingleTurnSample` 및 `EvaluationDataset` 클래스 가져옴

```
from ragas import SingleTurnSample, EvaluationDataset
```

    - **b.** **`개별 샘플 만들기`**: 개별 평가 샘플을 나타내는 `여러 개`의 `SingleTurnSample 인스턴스` 만들기

```
# Sample 1
sample1 = SingleTurnSample(
    user_input="What is the capital of Germany?",
    retrieved_contexts=["Berlin is the capital and largest city of Germany."],
    response="The capital of Germany is Berlin.",
    reference="Berlin",
)

# Sample 2
sample2 = SingleTurnSample(
    user_input="Who wrote 'Pride and Prejudice'?",
    retrieved_contexts=["'Pride and Prejudice' is a novel by Jane Austen."],
    response="'Pride and Prejudice' was written by Jane Austen.",
    reference="Jane Austen",
)

# Sample 3
sample3 = SingleTurnSample(
    user_input="What's the chemical formula for water?",
    retrieved_contexts=["Water has the chemical formula H2O."],
    response="The chemical formula for water is H2O.",
    reference="H2O",
)
```

    - **c.** **`EvaluationDataset 만들기`**: `SingleTurnSample` 인스턴스 목록 전달 → `EvaluationDataset` 만들기

```
dataset = EvaluationDataset(samples=[sample1, sample2, sample3])
```

- **`Hugging Face 데이터세트` → `평가 데이터세트 로드하기`**
  - `Hugging Face Datasets` 라이브러리와 같은 기존 데이터셋 소스에서 평가 데이터셋을 불러올 수 있음
  - 기존의 `Hugging Face 평가 데이터셋` 로드 → `EvaluationDataset` 인스턴스 변환하는 방법
    - 사용자 입력, 검색된 컨텍스트, 응답, 참조 등 평가에 필요한 필드가 데이터 세트에 포함되어 있는지 확인하기

```
from datasets import load_dataset

dataset = load_dataset("explodinggradients/amnesty_qa","english_v3")
```

    - `Ragas EvaluationDataset` 객체 = 데이터 세트 로드하기

```
from ragas import EvaluationDataset

eval_dataset = EvaluationDataset.from_hf_dataset(dataset["eval"])
```

- **`환경 설정`**

```
# API 키를 환경변수로 관리하기 위한 설정 파일
from dotenv import load_dotenv

# API 키 정보 로드
load_dotenv()                           # True
```

```
from langsmith import Client
from langsmith import traceable

import os

# LangSmith 환경 변수 확인

print("\n--- LangSmith 환경 변수 확인 ---")
langchain_tracing_v2 = os.getenv('LANGCHAIN_TRACING_V2')
langchain_project = os.getenv('LANGCHAIN_PROJECT')
langchain_api_key_status = "설정됨" if os.getenv('LANGCHAIN_API_KEY') else "설정되지 않음" # API 키 값은 직접 출력하지 않음

if langchain_tracing_v2 == "true" and os.getenv('LANGCHAIN_API_KEY') and langchain_project:
    print(f"✅ LangSmith 추적 활성화됨 (LANGCHAIN_TRACING_V2='{langchain_tracing_v2}')")
    print(f"✅ LangSmith 프로젝트: '{langchain_project}'")
    print(f"✅ LangSmith API Key: {langchain_api_key_status}")
    print("  -> 이제 LangSmith 대시보드에서 이 프로젝트를 확인해 보세요.")
else:
    print("❌ LangSmith 추적이 완전히 활성화되지 않았습니다. 다음을 확인하세요:")
    if langchain_tracing_v2 != "true":
        print(f"  - LANGCHAIN_TRACING_V2가 'true'로 설정되어 있지 않습니다 (현재: '{langchain_tracing_v2}').")
    if not os.getenv('LANGCHAIN_API_KEY'):
        print("  - LANGCHAIN_API_KEY가 설정되어 있지 않습니다.")
    if not langchain_project:
        print("  - LANGCHAIN_PROJECT가 설정되어 있지 않습니다.")
```

- 셀 출력

```
--- LangSmith 환경 변수 확인 ---
✅ LangSmith 추적 활성화됨 (LANGCHAIN_TRACING_V2='true')
✅ LangSmith 프로젝트: 'LangChain-prantice'
✅ LangSmith API Key: 설정됨
-> 이제 LangSmith 대시보드에서 이 프로젝트를 확인해 보세요.
```

```
# 데이터 로드하기
import pandas as pd

df = pd.read_csv("../15_Evaluations/data/ragas_synthetic_dataset.csv")
df.head()
```

- **`df.head()`** - (`0.8s`)

- df.head()`

1) `googletrans를 활용한 번역`

- `DeepL API Key` 발급
  - `API` 발급에서 결제, `IP` 우회 등 여러 문제 발생
- **`goolgletrans`를 활용한 번역**
  - 사전에 `VS Code` 터미널에 설치할 것

```
pip install googletrans
```

```
# googletrans로 번역

import pandas as pd
from googletrans import Translator
from tqdm import tqdm
import time

# Translator 생성
translator = Translator()
print("✅ googletrans 번역 시작")
```

- **`✅ googletrans 번역 시작`** - (`1.3s`)

```
# DataFrame 로드
df = pd.read_csv("data/ragas_synthetic_dataset.csv")
```

```
print(df.head())
```

- **`print(df.head())`**

```
user_input  \
0  미국 바이든 대통령은 안전하고 신뢰할 수 있는 AI 개발과 사용에 관한 행정명령을 ...
1  AI의 안전과 보안 기준 마련과 관련하여, 미국 정부에 정보 공유를 요구받는 컴퓨팅...
2  미국 행정명령은 AI가 사회 전반에 미치는 영향을 고려하여 어떤 분야에서 형평성과 ...
3            G7이 AI 기업을 대상으로 마련한 국제 행동강령의 명칭은 무엇인가요?
4    AI 국제 행동강령에서 AI 수명주기 전반에 걸쳐 요구하는 주요 조치들은 무엇인가요?

                                    reference  \
0                                  2023년 10월 30일
1  단일 데이터센터에서 1,000Gbit/s 이상의 네트워킹으로 연결되며 AI 훈련에서...
2  법률, 주택, 보건 분야에서 AI의 무책임한 사용으로 인한 차별과 편견 및 기타 문...
3  AI 국제 행동강령(International Code of Conduct for A...
4  AI 수명주기 전반에 걸쳐 위험을 평가 및 완화하고, 출시 및 배포 이후 취약점, ...

                          reference_contexts synthesizer_name
0  ['미국 바이든 대통령이 2023년 10월 30일 연방정부 차원에서 안전하고 신뢰할...           simple
1  ['△1026 플롭스(FLOPS, Floating Point Operation Pe...        reasoning
2  ['행정명령은 △AI의 안전과 보안 기준 마련 △개인정보보호 △형평성과 시민권 향상...    multi_context
3  ['G7이 첨단 AI 시스템을 개발하는 기업을 대상으로 AI 위험 식별과 완화를 위...           simple
4  ['행동강령은 AI 수명주기 전반에 걸친 위험 평가와 완화, 투명성과 책임성의 보장...        reasoning
```

```
# 데이터프레임의 컬럼명 출력
print(df.columns)
```

- **`print(df.columns)`**

```
Index(['user_input', 'reference', 'reference_contexts', 'synthesizer_name'], dtype='object')
```

```
# 컬럼 확인

print("✅ 현재 컬럼명:")
print(df.columns.tolist())
print(f"\n✅ 데이터 수: {len(df)}개")
print("\n샘플 데이터:")
print(df.head())
```

- ✅ 현재 컬럼명:

```
['user_input', 'reference', 'reference_contexts', 'synthesizer_name']
```

- ✅ 데이터 수: 9개

```
샘플 데이터:
                                          user_input  \
0  미국 바이든 대통령은 안전하고 신뢰할 수 있는 AI 개발과 사용에 관한 행정명령을 ...
1  AI의 안전과 보안 기준 마련과 관련하여, 미국 정부에 정보 공유를 요구받는 컴퓨팅...
2  미국 행정명령은 AI가 사회 전반에 미치는 영향을 고려하여 어떤 분야에서 형평성과 ...
3            G7이 AI 기업을 대상으로 마련한 국제 행동강령의 명칭은 무엇인가요?
4    AI 국제 행동강령에서 AI 수명주기 전반에 걸쳐 요구하는 주요 조치들은 무엇인가요?

                                          reference  \
0                                       2023년 10월 30일
1  단일 데이터센터에서 1,000Gbit/s 이상의 네트워킹으로 연결되며 AI 훈련에서...
2  법률, 주택, 보건 분야에서 AI의 무책임한 사용으로 인한 차별과 편견 및 기타 문...
3  AI 국제 행동강령(International Code of Conduct for A...
4  AI 수명주기 전반에 걸쳐 위험을 평가 및 완화하고, 출시 및 배포 이후 취약점, ...

                                 reference_contexts synthesizer_name
0  ['미국 바이든 대통령이 2023년 10월 30일 연방정부 차원에서 안전하고 신뢰할...           simple
1  ['△1026 플롭스(FLOPS, Floating Point Operation Pe...        reasoning
2  ['행정명령은 △AI의 안전과 보안 기준 마련 △개인정보보호 △형평성과 시민권 향상...    multi_context
3  ['G7이 첨단 AI 시스템을 개발하는 기업을 대상으로 AI 위험 식별과 완화를 위...           simple
4  ['행동강령은 AI 수명주기 전반에 걸친 위험 평가와 완화, 투명성과 책임성의 보장...        reasoning
```

```
# Translator 생성
translator = Translator()
print("✅ googletrans 3.1.0a0 번역 시작")
```

- ✅ **googletrans 3.1.0a0 번역 시작**

```
# 번역 함수
def translate_text(text):
    """googletrans로 영어 → 한국어 번역"""
    if pd.isna(text) or text == "":
        return text

    try:
        result = translator.translate(text, src='en', dest='ko')
        return result.text
    except Exception as e:
        print(f"⚠️ 번역 실패: {e}")
        time.sleep(1)
        try:
            result = translator.translate(text, src='en', dest='ko')
            return result.text
        except:
            return text
```

```
# 번역 실행
print("\n⏳ 번역 진행 중...")

for i, row in tqdm(df.iterrows(), total=len(df), desc="번역 진행 중"):
    # user_input 번역
    if pd.notna(row["user_input"]):
        df.loc[i, "user_input_translated"] = translate_text(row["user_input"])

    # reference 번역
    if pd.notna(row["reference"]):
        df.loc[i, "reference_translated"] = translate_text(row["reference"])

    time.sleep(0.2)

print("\n✅ 번역 완료!")
```

- `googletrans` - (`19.9s`)

```
⏳ 번역 진행 중...
번역 진행 중:   0%|          | 0/9 [00:00<?, ?it/s]⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
번역 진행 중:  11%|█         | 1/9 [00:02<00:17,  2.22s/it]⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
번역 진행 중:  22%|██        | 2/9 [00:04<00:15,  2.21s/it]⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
번역 진행 중:  33%|███       | 3/9 [00:06<00:13,  2.22s/it]⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
번역 진행 중:  44%|████      | 4/9 [00:08<00:11,  2.21s/it]⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
번역 진행 중:  56%|█████     | 5/9 [00:11<00:08,  2.21s/it]⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
번역 진행 중:  67%|██████    | 6/9 [00:13<00:06,  2.21s/it]⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
번역 진행 중:  78%|███████   | 7/9 [00:15<00:04,  2.21s/it]⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
번역 진행 중:  89%|████████  | 8/9 [00:17<00:02,  2.21s/it]⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
⚠️ 번역 실패: 'coroutine' object has no attribute 'text'
번역 진행 중: 100%|██████████| 9/9 [00:19<00:00,  2.22s/it]
✅ 번역 완료!
```

- 📊 **문제 분석**
  - `asynco`, `await` 기능 필요 → **번역 실패**
  - → `llm` 사용해서 번역하는 것으로 방향 바꾸기

---

2) `Groq으로 번역하기`

```
from langchain_groq import ChatGroq
from dotenv import load_dotenv
import os

from dotenv import load_dotenv
import os
import warnings
warnings.filterwarnings("ignore")               # 경고 메시지 무시

# API 키 확인 (GROQ_API_KEY로 변경)
if not os.getenv("GROQ_API_KEY"):        # 환경 변수에서 로드하거나 사용자에게 입력 요청
    api_key = os.getenv("GROQ_API_KEY")

    if not api_key:
        print("GROQ_API_KEY 환경 변수를 설정해주세요.")

# ChatGroq 사용
llm = ChatGroq(
    model="llama-3.1-8b-instant",
    temperature=0)                       # 0.1s
```

```
print("\n✅ Groq 번역 시작")

# 번역 함수
def translate_text(text):
    """Gemini로 영어 → 한국어 번역"""
    if pd.isna(text) or text == "":
        return text

    try:
        prompt = f"다음 영어 문장을 자연스러운 한국어로 번역하세요. 번역 결과만 출력하세요:\n\n{text}"
        result = llm.invoke(prompt)
        return result.content.strip()
    except Exception as e:
        print(f"⚠️ 번역 실패: {e}")
        return text
```

- ✅ **Groq 번역 시작**

```
# 번역 실행
print("\n⏳ 번역 진행 중...")

for i, row in tqdm(df.iterrows(), total=len(df), desc="번역 진행 중"):
    # user_input 번역
    if pd.notna(row["user_input"]):
        translated = translate_text(row["user_input"])
        df.loc[i, "user_input_translated"] = translated

    # reference 번역
    if pd.notna(row["reference"]):
        translated = translate_text(row["reference"])
        df.loc[i, "reference_translated"] = translated

print("\n✅ 번역 완료!")
```

- **번역 완료** - (5.8s)

```
⏳ 번역 진행 중...
번역 진행 중: 100%|██████████| 9/9 [00:05<00:00,  1.54it/s]
✅ 번역 완료!
```

```
# 번역 전 확인
print("\n📋 번역 전:")
print(df[['user_input', 'reference']].head())
```

- **번역 전**

```
📋 번역 전:
                                          user_input  \
0  미국 바이든 대통령은 안전하고 신뢰할 수 있는 AI 개발과 사용에 관한 행정명령을 ...
1  AI의 안전과 보안 기준 마련과 관련하여, 미국 정부에 정보 공유를 요구받는 컴퓨팅...
2  미국 행정명령은 AI가 사회 전반에 미치는 영향을 고려하여 어떤 분야에서 형평성과 ...
3            G7이 AI 기업을 대상으로 마련한 국제 행동강령의 명칭은 무엇인가요?
4    AI 국제 행동강령에서 AI 수명주기 전반에 걸쳐 요구하는 주요 조치들은 무엇인가요?

                                     reference
0                                   2023년 10월 30일
1  단일 데이터센터에서 1,000Gbit/s 이상의 네트워킹으로 연결되며 AI 훈련에서...
2  법률, 주택, 보건 분야에서 AI의 무책임한 사용으로 인한 차별과 편견 및 기타 문...
3  AI 국제 행동강령(International Code of Conduct for A...
4  AI 수명주기 전반에 걸쳐 위험을 평가 및 완화하고, 출시 및 배포 이후 취약점, ...
```

```
# 번역 후 확인
print("\n📋 번역 후:")
print(df[['user_input_translated', 'reference_translated']].head())
```

- **번역 후**

```
📋 번역 후:
                          user_input_translated  \
0  미국 바이든 대통령은 안전하고 신뢰할 수 있는 AI 개발과 사용에 관한 행정명령을 ...
1  AI의 안전과 보안 기준 마련과 관련하여, 미국 정부에 정보 공유를 요구받는 컴퓨팅...
2  미국 행정명령은 AI가 사회 전반에 미치는 영향을 고려하여 어떤 분야에서 형평성과 ...
3  G7이 AI 기업을 대상으로 마련한 국제 행동강령의 명칭은 'AI 비즈니스 코드'입니다.
4  AI 국제 행동강령에서 AI 수명주기 전반에 걸쳐 요구하는 주요 조치들은 AI 개발...

                             reference_translated
0                                   2023년 10월 30일
1  단일 데이터센터에서 1,000Gbit/s 이상의 네트워킹으로 연결되며 AI 훈련에서...
2  법률, 주택, 보건 분야에서 AI의 무책임한 사용으로 인한 차별과 편견 및 기타 문...
3                        고급 인공지능 시스템에 대한 국제 행동강령입니다.
4  AI 수명주기 전반에 걸쳐 위험을 평가 및 완화하고, 출시 및 배포 이후 취약점, ...
```

```
# 컬럼 정리
df.drop(columns=["user_input", "reference"], inplace=True)

df.rename(
    columns={
        "user_input_translated": "user_input",
        "reference_translated": "reference",
    },
    inplace=True,
)
```

```
# 저장
df.to_csv("../15_Evaluations/data/ragas_synthetic_dataset_translated.csv", index=False)

print("\n✅ 저장 완료!")
print(f"파일 경로: data/ragas_synthetic_dataset_translated.csv")
print("\n최종 데이터:")
print(df[['user_input', 'reference']].head())
```

- **저장 완료**

```
✅ 저장 완료!
파일 경로: data/ragas_synthetic_dataset_translated.csv

최종 데이터:
                                          user_input  \
0  미국 바이든 대통령은 안전하고 신뢰할 수 있는 AI 개발과 사용에 관한 행정명령을 ...
1  AI의 안전과 보안 기준 마련과 관련하여, 미국 정부에 정보 공유를 요구받는 컴퓨팅...
2  미국 행정명령은 AI가 사회 전반에 미치는 영향을 고려하여 어떤 분야에서 형평성과 ...
3  G7이 AI 기업을 대상으로 마련한 국제 행동강령의 명칭은 'AI 비즈니스 코드'입니다.
4  AI 국제 행동강령에서 AI 수명주기 전반에 걸쳐 요구하는 주요 조치들은 AI 개발...

                                          reference
0                                       2023년 10월 30일
1  단일 데이터센터에서 1,000Gbit/s 이상의 네트워킹으로 연결되며 AI 훈련에서...
2  법률, 주택, 보건 분야에서 AI의 무책임한 사용으로 인한 차별과 편견 및 기타 문...
3                          고급 인공지능 시스템에 대한 국제 행동강령입니다.
4  AI 수명주기 전반에 걸쳐 위험을 평가 및 완화하고, 출시 및 배포 이후 취약점, ...
```

---

4) Hugging Face Dataset 업로드

- **Pandas DataFrame → Hugging Face Dataset으로 변환 → 업로드 진행해보기**

```
from datasets import Dataset

# pandas DataFrame을 Hugging Face Dataset으로 변환
dataset = Dataset.from_pandas(df)

# 데이터셋 확인
print(dataset)
```

- **pandas DataFrame → Hugging Face Dataset**

```
Dataset({
    features: ['reference_contexts', 'synthesizer_name', 'user_input', 'reference', 'user_input_translated',
    num_rows: 9
})
```

```
print(type(dataset))        # <class 'datasets.arrow_dataset.Dataset'>
```

```
# 번역된 데이터 확인하기
df_upload = df.copy()

print("\n✅ 업로드할 데이터:")
print(df_upload.head())
```

- **업로드할 데이터로 copy하기**

```
✅ 업로드할 데이터:
                              reference_contexts synthesizer_name  \
0  ['미국 바이든 대통령이 2023년 10월 30일 연방정부 차원에서 안전하고 신뢰할...           simple
1  ['△1026 플롭스(FLOPS, Floating Point Operation Pe...        reasoning
2  ['행정명령은 △AI의 안전과 보안 기준 마련 △개인정보보호 △형평성과 시민권 향상...    multi_context
3  ['G7이 첨단 AI 시스템을 개발하는 기업을 대상으로 AI 위험 식별과 완화를 위...           simple
4  ['행동강령은 AI 수명주기 전반에 걸친 위험 평가와 완화, 투명성과 책임성의 보장...        reasoning

                                      user_input  \
0  미국 바이든 대통령은 안전하고 신뢰할 수 있는 AI 개발과 사용에 관한 행정명령을 ...
1  AI의 안전과 보안 기준 마련과 관련하여, 미국 정부에 정보 공유를 요구받는 컴퓨팅...
2  미국 행정명령은 AI가 사회 전반에 미치는 영향을 고려하여 어떤 분야에서 형평성과 ...
3  G7이 AI 기업을 대상으로 마련한 국제 행동강령의 명칭은 'AI 비즈니스 코드'입니다.
4  AI 국제 행동강령에서 AI 수명주기 전반에 걸쳐 요구하는 주요 조치들은 AI 개발...

                                      reference  \
0                                    2023년 10월 30일
1  단일 데이터센터에서 1,000Gbit/s 이상의 네트워킹으로 연결되며 AI 훈련에서...
2  법률, 주택, 보건 분야에서 AI의 무책임한 사용으로 인한 차별과 편견 및 기타 문...
3                          고급 인공지능 시스템에 대한 국제 행동강령입니다.
4  AI 수명주기 전반에 걸쳐 위험을 평가 및 완화하고, 출시 및 배포 이후 취약점, ...

                           user_input_translated  \
0  미국 바이든 대통령은 안전하고 신뢰할 수 있는 AI 개발과 사용에 관한 행정명령을 ...
1  AI의 안전과 보안 기준 마련과 관련하여, 미국 정부에 정보 공유를 요구받는 컴퓨팅...
2  미국 행정명령은 AI가 사회 전반에 미치는 영향을 고려하여 어떤 분야에서 형평성과 ...
3  G7이 AI 기업을 대상으로 마련한 국제 행동강령의 명칭은 'AI 비즈니스 코드'입니다.
4  AI 국제 행동강령에서 AI 수명주기 전반에 걸쳐 요구하는 주요 조치들은 AI 개발...

                                reference_translated
0                                    2023년 10월 30일
1  단일 데이터센터에서 1,000Gbit/s 이상의 네트워킹으로 연결되며 AI 훈련에서...
2  법률, 주택, 보건 분야에서 AI의 무책임한 사용으로 인한 차별과 편견 및 기타 문...
3                          고급 인공지능 시스템에 대한 국제 행동강령입니다.
4  AI 수명주기 전반에 걸쳐 위험을 평가 및 완화하고, 출시 및 배포 이후 취약점, ...
```

```
# HuggingFace Dataset으로 변환
dataset = Dataset.from_pandas(df_upload)

print("\n✅ Dataset 변환 완료:")
print(dataset)
print(f"\n특징: {dataset.features}")
print(f"데이터 수: {dataset.num_rows}개")
```

- ✅ Dataset 변환 완료:

```
Dataset({
    features: ['reference_contexts', 'synthesizer_name', 'user_input', 'reference', 'user_input_translated', 'ref
    num_rows: 9
})
```

- 특징:

```
{'reference_contexts': Value('string'), 'synthesizer_name': Value('string'), 'user_input': Value('string'), 'refe
```

- 데이터 수: 9개

```
# ========================================
# HuggingFace CLI 로그인 (권장!)
# ========================================
import os
```

```
from dotenv import load_dotenv

load_dotenv()

# HuggingFace CLI 로그인
from huggingface_hub import login

# .env에서 토큰 가져오기
token = os.getenv("HUGGINGFACEHUB_API_TOKEN")

print(f"토큰: {token[:4]}..." if token else "❌ 토큰 없음")

# 로그인 시도
try:
    login(token=token)
    print("✅ HuggingFace 로그인 성공!")
except Exception as e:
    print(f"❌ 로그인 실패: {e}")
```

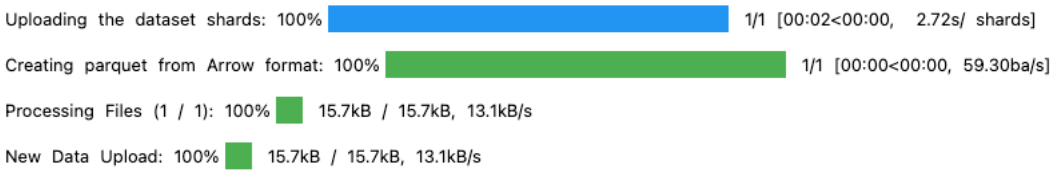
- 토큰: hf_c...
- ✅ HuggingFace 로그인 성공!

```
# ========================================
# 토큰 직접 확인
# ========================================
import os
from dotenv import load_dotenv

load_dotenv()

token = os.getenv("HUGGINGFACEHUB_API_TOKEN")

print("✅ 토큰 확인:")
print(f"토큰 존재: {'예' if token else '아니오'}")
print(f"토큰 길이: {len(token) if token else 0}자")
print(f"토큰 앞부분: {token[:4] if token else '없음'}...")
print(f"토큰 타입: {type(token)}")

# 토큰이 비어있는지 확인
if not token or token == "" or token == "None":
    print("\n❌ 토큰이 비어있습니다!")
    print("\n해결 방법:")
    print("1. HuggingFace에서 새 토큰 발급")
    print("2. .env 파일에 다음과 같이 저장:")
    print("   HUGGINGFACEHUB_API_TOKEN=hf_xxxxxxxxxxxxx")
    print("3. 주피터 노트북 재시작 (커널 재시작)")
else:
    print("\n✅ 토큰이 정상적으로 로드되었습니다!")
```

- ✅ 토큰 확인:
- 토큰 존재: 예
- 토큰 길이: 37자
- 토큰 앞부분: `hf_c...`
- 토큰 타입: <class 'str'>
- ✅ 토큰이 정상적으로 로드되었습니다!

```
# HuggingFace Hub 업로드
dataset_name = "livemylife23/rag-synthetic-dataset-korean"

print(f"\n⏳ 업로드 시도: {dataset_name}")

print(f"\n⏳ HuggingFace Hub 업로드 중...")
print(f"데이터셋 이름: {dataset_name}")

try:
    dataset.push_to_hub(
        dataset_name,
        private=True,           # private=False로 설정 → 공개
        split="train",          # 또는 "korean_v1", "test" 등
        token=os.getenv("HUGGINGFACEHUB_API_TOKEN"),
    )

    print("\n✅ 업로드 완료!")
    print(f"\n🔗 URL: https://huggingface.co/datasets/{dataset_name}")
    print("\n📌 참고: Dataset Viewer는 표기되기까지 시간이 좀 걸릴 수 있습니다.")

except Exception as e:
```

```
    print(f"\n❌ 업로드 실패: {e}")
    print("\n확인 사항:")
    print("1. HUGGINGFACEHUB_API_TOKEN이 .env 파일에 있는지")
    print("2. 토큰이 Write 권한인지")
    print("3. 데이터셋 이름이 올바른지 (your_username/dataset_name)")
```

- ⏳ 업로드 시도: livemylife23/rag-synthetic-dataset-korean
- ⏳ HuggingFace Hub 업로드 중...
- 데이터셋 이름: livemylife23/rag-synthetic-dataset-korean
- 

- ✅ 업로드 완료!
- 🔗 URL: https://huggingface.co/datasets/livemylife23/rag-synthetic-dataset-korean
- 📌 *참고: Dataset Viewer는 표기되기까지 시간이 좀 걸릴 수 있습니다.*

- next: ***`04. LangSmith 데이터셋 생성`***
