

여행 일정 최적화 챗봇 서비스 RAG 설계 및 보고서

- 광주 4팀(황치운, 강지민, 김성훈, 윤상묵, 하지원)

RAG 파이프라인 구축 프로세스 보고서(안)	
RAG 파이프라인 설계	<p>데이터 최적화:</p> <ul style="list-style-type: none">• Chunk Size: 1000• Overlap: 100 <p>벡터 데이터베이스 구축 및 임베딩:</p> <ul style="list-style-type: none">• 벡터 DB : Pinecone• 임베딩 모델 : Upstage Solar Embedding <p>Retriever 및 Reranker 구현:</p> <ul style="list-style-type: none">• Retriever: Hybrid Retrieval System<ul style="list-style-type: none">○ Dense Retriever<ul style="list-style-type: none">▪ 하이퍼 파라미터 설정<ul style="list-style-type: none">▪ 반환할 문서 수 (k) = 4▪ 유사도 임계값 = 0.7▪ 장소 설명, 분위기, 여행 스타일 등 의미적 유사성 검색○ Sparse Retriever (키워드 기반)<ul style="list-style-type: none">▪ BM25 알고리즘 활용▪ 정확한 장소명, 주소, 영업시간 등 팩트 정보 검색• (optional) Reranker : CrossEncoder<ul style="list-style-type: none">○ 두 Retriever 결과를 결합하여 재순위화○ 상위 10개 문서로 필터링○ 고려 요소:<ul style="list-style-type: none">▪ 현재 시간대 영업 여부▪ 사용자 위치와의 거리▪ 이전 방문 장소와의 연계성 <p>LLM 프롬프트 설계 및 답변 생성, 평가 :</p> <p>1/ Task 정의 : 여행 일정 계획 챗봇</p> <p>2/ 프롬프트 설계 :</p>

SYSTEM_TEMPLATE = """당신은 전문적인 여행 플래너입니다.
주어진 정보를 기반으로 최적화된 여행 계획을 제안해주세요.

CONTEXT:
{context}

지켜야 할 규칙:

1. 시간대별 동선을 고려하여 효율적인 일정을 만드세요.
2. 각 장소의 운영시간과 혼잡도를 반영하세요.
3. 이동 수단과 소요 시간을 명시하세요.
4. 식사 시간대에는 인근 맛집을 추천하세요.

제공할 정보 형식:

1. 전체 일정 요약
2. 시간대별 상세 일정
| - HH:MM | 장소명 | 체류시간 | 이동수단 | 특이사항
3. 예상 소요 비용
4. 준비물 및 주의사항

사용자 선호도:

- 여행 스타일: {travel_style}
- 예산: {budget}
- 체력 수준: {energy_level}
- 식사 제약: {dietary_restrictions}

반드시 참고할 현재 상황:

- 현재 시각: {current_time}
- 현재 위치: {current_location}
- 혼잡도 정보: {crowdedness}

"""

User input template

USER_TEMPLATE = """여행 조건:

- 여행 일자: {travel_dates}
- 여행지: {destination}
- 동행인: {companions}
- 선호 활동: {preferred_activities}
- 제외할 것: {exclusions}

위 조건으로 여행 계획을 세워주세요."""

```

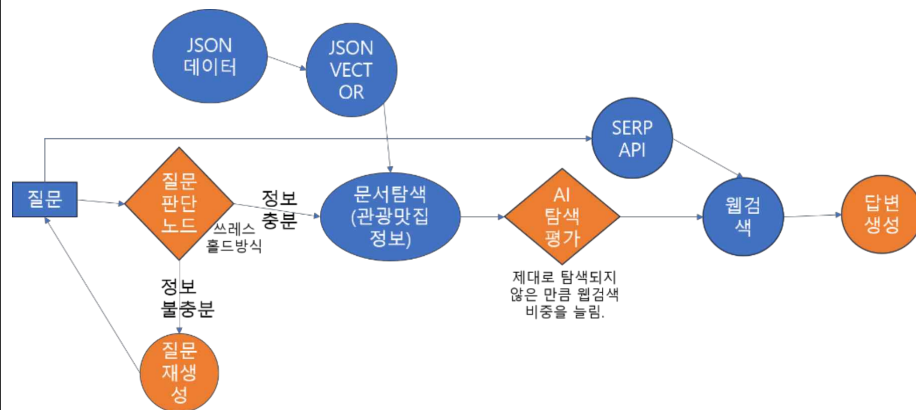
# Create prompt template
prompt = ChatPromptTemplate.from_messages([
    ("system", SYSTEM_TEMPLATE),
    ("human", USER_TEMPLATE)
])

# Example usage
example_variables = {
    # System context
    "context": "검색된 관련 여행 정보...",
    "travel_style": "여유롭게",
    "budget": "1인당 30만원",
    "energy_level": "보통",
    "dietary_restrictions": "없음",
    "current_time": "09:00",
    "current_location": "서울역",
    "crowdedness": "보통",

    # User input
    "travel_dates": "2024-12-31~2025-01-01",
    "destination": "부산",
    "companions": "친구 2명",
    "preferred_activities": ["맛집", "야경", "카페"],
    "exclusions": ["과격한 활동", "산행"]
}

```

3/ LangGraph 설계 :



4/ 답변 생성 모델 : Upstage Solar Pro

5/ 답변 사후 평가 :

- Upstage Groundness Check 으로 Hallucination 평가
- 실제 운영 시간, 거리, 교통 정보 정확성 확인

RAG 파이프라인 평가 및 결과

평가방법

정량 평가 : RAGAS 평가 지표

- **context_precision:** 검색한 문서 중에서 진짜로 관련된 문서가 차지하는 비율
 - 추천된 호텔, 교통편, 레스토랑, 명소 정보 중 실제로 관련된 정보의 비율

- **context_recall:** 실제로 관련된 문서 중에서 얼마나 많이 검색에 성공했는지
 - 실제 관련 정보 중 추천에서 얼마나 많이 검색되었는지
- **faithfulness:** 생성된 답변이 가지고 있는 지식으로 얼마나 뒷받침 되는지에 대한 비율
 - 추천된 일정이 검색된 정보와 얼마나 일치하고 신뢰할 수 있는지
- **answer_relevancy:** 생성된 답변이 주어진 질문과 얼마나 관련성이 있는 지

```
from datasets import Dataset
import os
from ragas import evaluate
from ragas.metrics import answer_relevancy, context_precision, faithfulness, context_recall, answer_correctness

# OpenAI API Key 설정
os.environ["OPENAI_API_KEY"] = "sk-"

# 한국 여행 일정 데이터를 생성
travel_data_samples = {
    'question': [
        '서울에서 3일 동안 머물기에 가장 좋은 호텔은 어디인가요?',
        '서울에서 부산까지 가장 저렴한 교통편은 무엇인가요?',
        '서울에서 저녁 식사를 하기 좋은 상위 3개의 레스토랑은 어디인가요?',
        '서울에서 짧은 여행 동안 꼭 방문해야 할 장소는 어디인가요?'
    ],
    'answer': [
        '롯데호텔 서울은 매우 추천되는 호텔입니다.',
        '가장 저렴한 교통편은 고속버스로, 비용은 약 15,000원입니다.',
        '한복마을, 광화문 미쉐린 레스토랑, 그리고 남산타워 레스토랑이 상위권에 있습니다.',
        '경복궁, 한강공원, 그리고 남산타워는 꼭 방문해야 할 장소입니다.'
    ],
    'contexts': [
        '롯데호텔 서울은 최고급 편의시설을 제공하며, 중심부에 위치해 있습니다.',
        '서울에서 부산까지의 고속버스 요금은 약 15,000원부터 시작합니다.',
        '한복마을은 전통 한식을 즐길 수 있는 유명 레스토랑입니다.',
        '광화문 미쉐린 레스토랑은 고급스러운 분위기와 맛으로 유명합니다.',
        '남산타워 레스토랑은 아름다운 서울 전경을 제공합니다.',
        '경복궁, 한강공원, 남산타워는 꼭 방문해야 할 장소입니다.'
    ],
    'ground_truth': [
        '롯데호텔 서울은 최고의 호텔 중 하나입니다.',
        '고속버스는 서울에서 부산까지 가장 저렴한 교통편입니다.',
        '한복마을, 광화문 미쉐린 레스토랑, 남산타워 레스토랑이 상위 3곳입니다.',
        '경복궁, 한강공원, 남산타워는 꼭 방문해야 할 장소입니다.'
    ]
}

# 데이터셋 생성
dataset = Dataset.from_dict(travel_data_samples)

score = evaluate(dataset, metrics=[
    answer_relevancy,
    context_precision,
    faithfulness,
    context_recall,
    answer_correctness
])

# 평가 결과 출력
score_df = score.to_pandas()
print(score_df)
```

- 제공된 일정이 사용자의 요구를 얼마나 관련성을 충족시키는지

정성 평가

- **샘플링 방식**
 - 무작위로 10개의 질문을 선택하여 챗봇의 답변을 평가합니다.
 - 질문은 제품 매뉴얼의 다양한 섹션에서 추출된 내용을 기반으로 구성합니다.
- **평가 항목**
 - **정확성:** 생성된 답변이 제품 매뉴얼의 내용과 얼마나 일치하는가?
 - **관련성:** 답변이 검색된 문서/데이터와 관련이 있는가?
 - **명확성:** 답변이 쉽게 이해되고 논리적으로 명확한가?
- **평가 절차**
 - 각 질문에 대해 생성된 답변을 매뉴얼의 실제 내용을 기준으로 비교 검토합니다.
 - 관련성이 낮거나 잘못된 답변은 피드백을 기록하여 개선 방안을 도출합니다

<div>결론 및 향후 발전 방향</div>	<div><h3>결론</h3><p>여행 일정 최적화 AI 챗봇 서비스는 사용자에게 맞춤형 여행 일정을 제공함으로써 개인의 여행 경험을 극대화할 수 있습니다. 이를 위해 벡터 DB를 구축하고, 이를 바탕으로 LLM을 활용한 RAG(Relevant Augmented Generation) 파이프라인을 설계하여 공공데이터 및 카카오톡 API를 기반으로 신속하고 정확한 답변을 제공합니다.</p><p>본 서비스를 통해 사용자는 효율적인 일정 관리를 경험하고, 서비스 제공자는 사용자 피드백과 데이터를 활용하여 지속적으로 서비스를 개선하며 관광 산업 발전에 기여할 수 있습니다.</p></div> <div><h3>향후 발전 방향</h3><div><div>- 다국어 지원 확대</div><p>글로벌 사용자 대상으로 서비스 확장을 위해 다국어 지원 기능을 추가합니다.</p><p>해외 관광지 추천 사이트 및 음식점 리뷰 데이터를 수집하여 서비스의 국제화 가능성을 높입니다.</p></div><div><div>- 멀티턴 대화 기능 추가</div><p>사용자의 복잡한 요청이나 추가 질문을 처리할 수 있도록 멀티턴 대화 기능을 강화합니다.</p><p>지속적인 대화 흐름을 통해 사용자와 LLM 간의 인터랙션을 개선하여 더욱 세부적인 여행 계획을 지원합니다.</p></div><div><div>- 데이터 확장 및 업데이트</div><p>공공데이터뿐만 아니라 사용자 리뷰, 블로그 데이터, 현지 가이드 정보 등을 추가로 수집하여 데이터베이스를 확장합니다.</p><p>데이터 업데이트 주기를 단축하여 최신 정보를 바탕으로 더욱 신뢰성 높은 추천을 제공합니다.</p></div><div><div>- 서비스 사용자 경험 개선</div><p>사용자 인터페이스(UI) 및 사용자 경험(UX)을 지속적으로 개선하여 중장년층도 쉽게 사용할 수 있는 직관적인 디자인을 도입합니다.</p><p>체력 수준, 음식 선호도 등 사용자 입력 과정을 간소화하여 편리성을 높입니다.</p></div><div><div>- 지역 특화 서비스 제공</div></div></div>
------------------------------	---

	<p>특정 지역의 관광지나 맛집 정보를 더욱 심층적으로 제공하여 로컬 관광 활성화에 기여합니다.</p> <p>지역 데이터 분석을 기반으로 새로운 관광 코스를 제안하거나 숨은 명소를 발굴하는 기능을 추가합니다.</p>
--	--