

# WITY

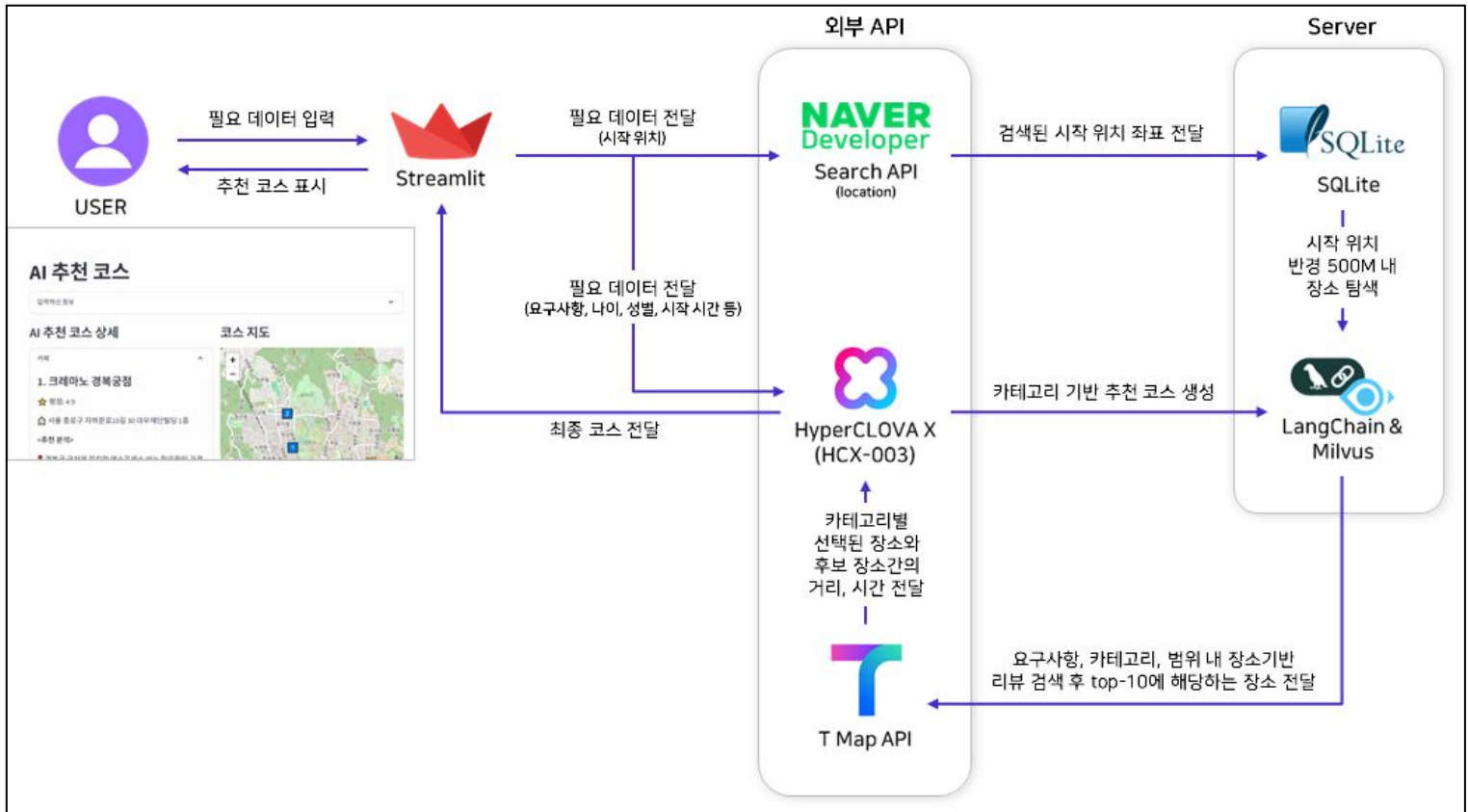
## What should I do Today?

본 프로젝트는 기존 장소 추천 서비스의 한계를 보완하여, 자연어 형태의 사용자 요구사항 기반으로 개인 일정을 경로 형태로 제공하는 시스템을 목표로 한다. 기존의 서비스는 여행에 초점이 맞춰져 있어, 스케일이 크거나 데일리한 활용이 어려운 경우가 많았다. 또한, 개인화된 추천을 위해 개인 로그 데이터 기반의 추천 시스템이 많아, 즉각적인 요구사항 반영이 어려운 경우도 많았다.

본 프로젝트는 이러한 문제를 해결하여, 사용자 요구사항 반영 및 데일리한 활용에 특화된 경로 생성 서비스를 개발하고자 한다. 이를 통한 주요 기대 효과는 다음과 같다.

- 실시간으로 발생하는 사용자 요구사항 반영
  - ➔ 기존 서비스가 사용자 기록에 의존한 편리한 추천에 집중하는 것과 달리, 사용자의 개별적인 요구사항을 직접 입력 받아 매번 새로운 추천을 하는 것에 중점을 두고 있다.
- 일상에서의 계획 생성 간편화
  - ➔ 데일리한 계획 생성에서 활용할 수 있도록 하여, 장소를 직접 검색하고 리뷰를 하나하나 찾아보는 시간을 줄이는 등 고객 편리성을 추구한다.

이 시스템은 “1. 사용자 입력”, “2. 카테고리 기반 경로 생성”, “3. 카테고리 별 후보지 선정”, “4. 요구 사항 만족도 및 경로 최적화를 고려한 장소 선정”의 주요 단계를 거친다. 이를 위해, **Streamlit** 기반의 웹 인터페이스·외부 **API**·**AI** 모델·데이터베이스 및 검색 엔진 등을 유기적으로 연결하여 동작하도록 설계했다.



[그림 num] 서비스 아키텍처

# 데이터 수집 및 전처리

## 데이터 수집

데이터는 Naver, Kakao, Google map에서 장소 정보와 리뷰를 웹 스크래핑하여 수집하였다. 음식점 데이터와 명소 데이터를 따로 스크래핑 하였는데 방법은 다음과 같다.

음식점 데이터 웹 스크래핑	명소 데이터 웹 스크래핑
LOCALDATA(지방행정 인허가 데이터개방) 사이트에서 일반음식점 데이터의 음식점 리스트 및 위치 정보 이용하여, “서울특별시 종로구” 기준 <b>6605개</b> 의 음식점 정보와 리뷰 데이터( <b>30개</b> 씩)를 스크래핑 하였다.	명소 데이터는 Naver map에서 “종로구 명소“로 검색하여 <b>377개</b> 의 명소에 대해 스크래핑 하고, 이 데이터를 이용하여, 다시 Kakao, Google map에서 스크래핑하여 명소 정보와 리뷰 데이터( <b>30개</b> 씩)를 수집하였다.

## 데이터 전처리

### 장소(음식점+명소) 정보 데이터 전처리

1. 검색 결과가 없을 때 비슷한 결과를 보여주는 데, 이 때 다른 지역의 장소가 나올 때가 있음  
→ 우리의 타겟 지역이 아닌 장소 데이터 삭제
2. 상호명 변경, 폐업 등의 이유로 Naver, Kakao, Google map에서 검색했을 때, 장소명이 다른 경우  
→ 후 작업에서 Naver API를 사용할 예정이기 때문에 Naver에서 검색했을 때, 결과가 나온 장소만 사용
3. 컬럼, 주소, 영업시간 등 형식 통일 → Naver 기준으로 통일
4. 리뷰 데이터가 없는 장소 데이터 삭제
5. 별점 → Naver, Kakao, Google map 별점 평균

### 리뷰(음식점+명소) 데이터 전처리

양질의 리뷰 데이터 확보를 위해 전처리 진행

1. 이모지 제거
2. “:”)와 같은 이모티콘이나 너무 많은 마침표, 느낌표 등을 제거하기 위해 모든 특수문자 제거
3. 하나의 리뷰가 숫자, 특수문자, 한글 자음, 모음으로만 이루어진 경우 제거
4. 음식점 리뷰가 5글자 이하인 경우, 명소 리뷰가 3글자 이하인 경우 제거
5. 위의 리뷰 전처리 과정에서 발생한 리뷰가 없는 장소는 삭제 처리

## 최종 데이터

데이터는 4,092개의 장소(음식점+명소) 정보 데이터와 154,652개의 해당 장소 리뷰 데이터로 구성하였다.

	장소 정보 데이터	리뷰 데이터
구성	id, 장소명, 대분류, 소분류, 주소, 영업시간, 별점, 인당가격	id, domain, 장소명, 리뷰
대분류	음식점, 카페, 주점	공연전시, 공원, 시장, 체험관광
형태	(4092,8)	(154652,4)

[표 num] 최종 데이터

# Search API활용 좌표 전달 및 반경내 장소 탐색

사용자가 가고자하는 장소를 입력하면 해당하는 장소가 정확하게 검색이되거나 근처로 검색이 되어야한다. 또한, 본 프로젝트에서는 사용자가 도보거리로 이동할 수 있는 경로로 고려해야하기 때문에 해당 장소 기준으로 반경 500m ~ 1km 이내의 장소들만 Retrieval 모델에 제공되어야한다.

## 수집된 데이터에 위/경도 값

추가  
수집된 스크래핑 데이터에서는 WGS84 좌표계 값인 위도(Latitude)와 경도(Longitude)가 존재하지 않는다. 사용자 입력 장소에서 반경 내 장소들을 탐색하기 위해서는 위/경도 값이 필요하므로 Naver Map API 중 Geocoding API를 사용하여 주소를 geocode로 변환하여 각 컬럼으로 추가하였다.

URI	https://naveropenapi.apigw.ntruss.com/map-geocode/v2/geocode
Query	address (ex. "서울 종로구 창경궁로 246 1, 2층")
Return	{“x”:longitude, “y”:latitude}

[표 num] Geocoding Request 값

## 시작 장소에 대한 탐색

geopy 라이브러리는 간단하게 pip로 설치 가능하며, Nominatim에서 제공하는 OpenStreetMap 데이터를 기반으로 주소의 위/경도를 구할 수 있다. 또한, 무료이며 사용량에 제한이 없지만 초당 Request를 약 1개씩 처리하므로 서비스에는 적합하지 않다.

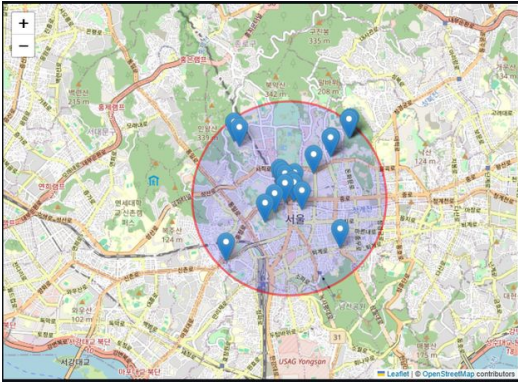
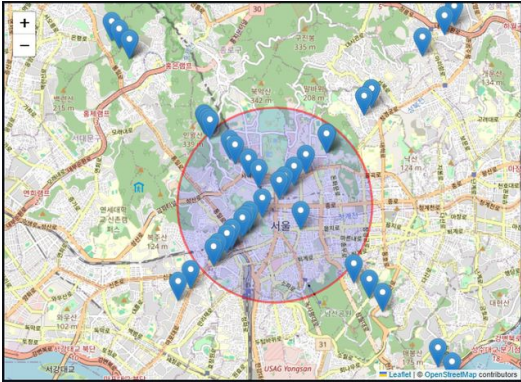
Naver Search(지역) API는 실제 네이버 지도에서 검색한 것과 동일한 결과를 출력한다. Naver Search API는 무료이지만, 하루 사용 제한량(25,000회)이 존재한다. 프로토타입 단계에서는 하루 사용 제한량이 넘지 않으므로 본 프로젝트에서 해당 API를 채택하였다.

URI	https://openapi.naver.com/v1/search/local.json
Params	{“query”: 사용자 입력 장소, “display”: 검색 표시 개수, “sort”: random(가장 유사도 높은 장소)}
Return	{“title”: 실제 검색된 장소 이름, “mapx”: longitude, “mapy”: latitude}

[표 num] Search API Request 값

## 반경 내 장소 탐색

검색된 시작 장소에서부터 아크 코사인 함수를 이용한 Haversine 공식을 적용하여 각 장소들의 거리를 계산하여 필터링을 진행하였다.



```
SELECT b.id, b.name,
c.main_category, c.category,
l.address, l.latitude, l.longitude,
b.rating, b.price_per_one,
b.business_hours, (6371000 *
acos( cos(radians(?)) *
cos(radians(l.latitude)) *
cos(radians(l.longitude) -
radians(?)) + sin(radians(?)) *
sin(radians(l.latitude)) )) ...
```

[그림 num] 반경 내 마커 표시 (좌 : 전체 예시 마커, 중간 : 필터링된 마커, 우 : 사용된 쿼리)

# 카테고리 기반 경로 생성

## 카테고리 기반 경로 생성의 필요성 및 문제 정의

기존 서비스들은 리뷰 평점, 방문자 수 등의 정량적 요소를 기반으로 인기 있는 장소를 추천하거나, 고정된 루트를 일부 변경하는 방식으로 경로를 제공한다. 그러나 이러한 접근 방식은 사용자의 개별적인 요구 사항을 즉각적으로 반영하기 어렵다는 한계를 가진다.

본 프로젝트에서는 사용자의 요구 사항을 반영한 맞춤형 당일 일정 경로 추천을 목표로 한다. 이를 위해, 먼저 요구사항에 맞는 카테고리 기반 경로를 생성한 후, 각 카테고리에서 최적의 장소를 추천하는 방식을 채택하였다. 즉, '사용자의 요구 사항', '연령대', '성별', '일정 시작 시각'의 입력을 바탕으로 적절한 카테고리 기반 경로를 추천하는 것이 핵심이다.

## 프롬프팅(Prompting) 기반 경로 생성

### Why prompting?

- 지도 학습 방식(Supervised Fine-Tuning, SFT)의 한계  
경로 추천 모델을 학습하려면 (입력: 사용자의 요청, 출력: 추천 경로)와 같은  $x, y$  쌍의 데이터가 필요하다. 그러나 데이터를 구축하려면 대량의 사용자 피드백과 검증된 추천 경로가 필요해 비용과 시간이 많이 소요된다. 또한, 정확한 정답이 명확하지 않은 문제에서  $y$  값을 임의로 설정할 경우 다양성이 부족해질 우려가 있다.
- LLM의 지식 활용 및 비용 효율성  
LLM은 이미 방대한 범용 지식을 학습했기 때문에, 별도의 데이터 없이도 공간적 연관성과 사용자 요청의 의미를 이해하여 적절한 추천을 생성할 수 있다. 따라서 추가적인 학습 없이 프롬프트 설계만으로도 원하는 결과를 얻을 수 있어, 비용과 적용 속도 측면에서도 효율적이다.

### Few-shot

단순히 입력과 출력 예시를 주어 LLM이 정해진 출력 양식에 맞춰 추천 경로를 생성하도록 하였다.

### Chain-of-Thought(CoT)

LLM이 논리적인 사고 과정을 거쳐 더 자연스럽고 최적화된 경로를 생성할 수 있도록 유도한다. 즉, 단순한 카테고리 나열이 아니라, 사용자의 요구를 분석하고 적절한 카테고리를 순차적으로 선택하도록 유도하도록 한다.

### Self-Refine

LLM이 첫 번째 생성 결과를 검토하고, 이를 자체적으로 개선하여 최적의 결과를 도출하는 기법. 즉, 사용자 요구사항에 의해 1차 일정을 생성하고, 이를 평가 및 개선하는 방식으로 최종 경로를 생성하도록 유도한다.

CoT	<ol style="list-style-type: none"><li>사용자의 요구를 분석하여 핵심 키워드 추출 (감정 상태, 특별한 이벤트, 선호하는 환경, 신체적 제약 등)</li><li>주어진 카테고리에서 사용자의 요구사항과 가장 잘 맞는 카테고리를 선정</li><li>일정 시작 시각을 고려하여 자연스러운 흐름으로 경로를 배치</li><li>최종적으로 카테고리 순서만 출력</li></ol>
Self-Refine	<ol style="list-style-type: none"><li>1차 일정을 생성</li><li>1차 일정의 문제점을 분석. 만약 문제가 없다면 1차 일정을 사용.</li><li>분석한 문제점을 기반으로 개선 방향 제시</li><li>개선 방향을 바탕으로 일정을 최적화하여 최종 일정 출력.</li></ol>

[그림 num] CoT와 Self-Refine 과정 유도 프롬프팅

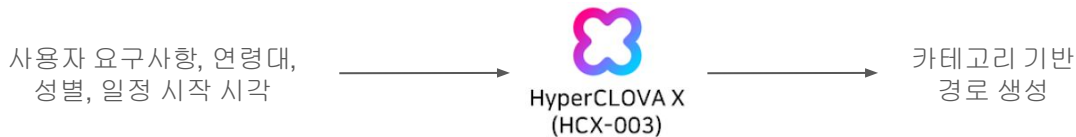
각 방법을 적용하여 생성된 결과 예시는 다음 페이지에 첨부하였다.

# 카테고리 기반 경로 생성

## 카테고리 기반 경로 생성의 필요성 및 문제 정의

기존 서비스들은 리뷰 평점, 방문자 수 등의 정량적 요소를 기반으로 인기 있는 장소를 추천하거나, 고정된 루트를 일부 변경하는 방식으로 경로를 제공한다. 그러나 이러한 접근 방식은 사용자의 개별적인 요구 사항을 즉각적으로 반영하기 어렵다는 한계를 가진다.

본 프로젝트에서는 사용자의 요구 사항을 반영한 맞춤형 당일 일정 경로 추천을 목표로 한다. 이를 위해, 먼저 요구사항에 맞는 카테고리 기반 경로를 생성한 후, 각 카테고리에서 최적의 장소를 추천하는 방식을 채택하였다. 즉, '사용자의 요구 사항', '연령대', '성별', '일정 시작 시각'의 입력을 바탕으로 적절한 카테고리 기반 경로를 추천하는 것이 핵심이다.



## 프롬프팅(Prompting) 기반 경로 생성

### Why prompting?

- 지도 학습 방식(Supervised Fine-Tuning, SFT)의 한계  
경로 추천 모델을 학습하려면 (입력: 사용자의 요청, 출력: 추천 경로)와 같은  $x, y$  쌍의 데이터가 필요하다. 그러나 데이터를 구축하려면 대량의 사용자 피드백과 검증된 추천 경로가 필요해 비용과 시간이 많이 소요된다. 또한, 정확한 정답이 명확하지 않은 문제에서  $y$  값을 임의로 설정할 경우 다양성이 부족해질 우려가 있다.
- LLM의 지식 활용 및 비용 효율성  
LLM은 이미 방대한 범용 지식을 학습했기 때문에, 별도의 데이터 없이도 공간적 연관성과 사용자 요청의 의미를 이해하여 적절한 추천을 생성할 수 있다. 따라서 추가적인 학습 없이 프롬프트 설계만으로도 원하는 결과를 얻을 수 있어, 비용과 적용 속도 측면에서도 효율적이다.

### Few-shot

단순히 입력과 출력 예시를 주어 LLM이 정해진 출력 양식에 맞춰 추천 경로를 생성하도록 하였다.

### Chain-of-Thought(CoT)

LLM이 논리적인 사고 과정을 거쳐 더 자연스럽게 최적화된 경로를 생성할 수 있도록 유도한다.

즉, 단순한 카테고리 나열이 아니라, 사용자의 요구를 분석하고 적절한 카테고리를 순차적으로 선택하도록 유도하도록 한다.

1. 사용자의 요구를 분석하여 핵심 키워드 추출  
(감정 상태, 특별한 이벤트, 선호하는 환경, 신체적 제약 등)
2. 주어진 카테고리에서 사용자의 요구사항과 가장 잘 맞는 카테고리를 선정
3. 일정 시작 시각을 고려하여 자연스러운 흐름으로 경로를 배치
4. 최종적으로 카테고리 순서만 출력

[그림 num] CoT 과정 유도 프롬프팅

### Self-Refine

LLM이 첫 번째 생성 결과를 검토하고, 이를 자체적으로 개선하여 최적의 결과를 도출하는 기법.

즉, 사용자 요구사항에 의해 1차 일정을 생성하고, 이를 평가 및 개선하는 방식으로 최종 경로를 생성하도록 유도한다.

1. 1차 일정을 생성
2. 1차 일정의 문제점을 분석. 만약 문제가 없다면 1차 일정을 사용.
3. 분석한 문제점을 기반으로 개선 방향 제시
4. 개선 방향을 바탕으로 일정을 최적화하여 최종 일정 출력.

[그림 num] Self-Refine 과정 유도 프롬프팅



# 카테고리 기반 경로 생성

## 각 프롬프팅 방법에 따른 생성 결과

‘Few Shot’, ‘Chain-of-Thought(CoT)’, ‘Self-Refine’ 방법에 따른 생성 결과 예시이다.  
같은 상황에 대해 각 프롬프팅 방법에 따라 카테고리 기반 경로가 어떻게 생성되는지에 대한 예시이며,  
‘CoT’와 ‘Self-Refine’의 경우는 사고 과정도 함께 첨부하였다.

	Few Shot	CoT	Self-Refine
사용자 요구사항	요즘 회사일로 스트레스가 심한데 친구 만나서 수다도 떨고 힐링 할 수 있는 방법 있을까요?		
성별	여성		
연령대	30대		
일정 시작 시각	17시		
생성 경로	공원 → 음식점 → 카페	주점 → 카페 → 공원	카페 → 음식점 → 공원 → 주점

[그림 num] 각 방법에 따른 생성 결과

	CoT	Self-Refine
경로 생성 이유	<div>1. 스트레스 해소를 위해서는 마음 놓고 이야기 할 수 있는 공간과 가벼운 활동이 필요함.</div> <div>2. 저녁 시간이므로 식사 겸 가벼운 음주를 할 수 있는 주점이 적절함.</div> <div>3. 이후에는 조용하게 대화를 나눌 수 있는 카페가 적당함.</div> <div>4. 마지막으로 밤거리를 산책하면서 생각을 정리하면 스트레스 해소에 도움이 될 것임.</div>	<div>1. 1차 일정: 카페 → 음식점 → 주점</div> <div>2. 1차 일정의 문제점 분석<ul style="list-style-type: none"><li>- 식사와 음주 위주의 일정으로 구성되어 있어 휴식이나 문화생활 등 다른 측면에서의 힐링 요소가 부족함.</li><li>- 서로 이야기를 나눌 수 있는 공간이지만 좀 더 편안하게 즐길 수 있는 분위기 조성이 필요해 보임.</li></ul></div> <div>3. 개선방향<ul style="list-style-type: none"><li>- 조용한 분위기에서 책을 읽거나 차를 마시며 여유로운 시간을 보낼 수 있는 장소를 추가하면 좋을 것 같음.</li><li>- 가벼운 운동이나 산책을 통해 몸과 마음을 이완시키는 것도 좋은 방법임.</li></ul></div>

[그림 num] CoT와 Self-Refine 사고 과정

# 조건에 맞는 리뷰 검색 및 해당 장소 전달

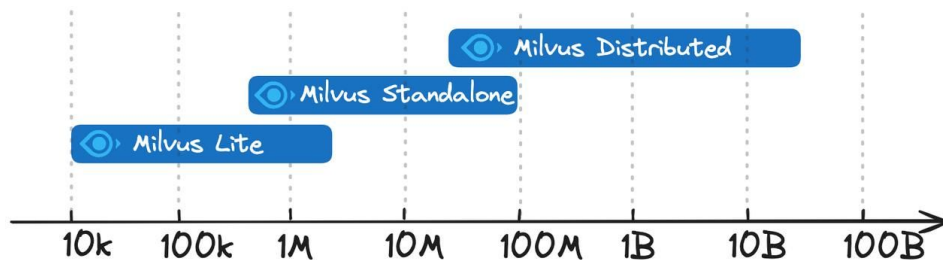
## Retrieval

### VectorDB 구성

Field	Data Type	Index	Remarks
Id	INT64	-	장소 ID
name	VARCHAR	-	장소명
sub_category	VARCHAR	-	소분류
text	VARCHAR	-	요약된 리뷰
positive_text	VARCHAR	-	요약된 리뷰 (긍정적 내용)
dense_vector	FLOAT_VECTOR	FLAT	Clova Studio embedding API (clir-emb-dolphin)
sparse_vector	FLOAT_VECTOR	SPARSE_INVERTED_INDEX	BM25 / k1=1.5, b=0.75, epsilon=0.25

[표 num] vectorDB filed 구성 정보

사용자 요구사항과 장소 리뷰 간의 유사도 기반 검색을 수행하기 위해, Milvus의 vectorDB를 활용했다. 특정 장소에 한정된 검색을 수행해야 했기 때문에 filtering을 지원하는 Milvus를 선택했고, 4,092개 정도로 데이터의 사이즈가 크지 않았기 때문에 로컬에서 쉽게 활용하고 관리할 수 있는 Milvus Lite를 활용했다. VectorDB를 구성하는 collection은 카테고리 별 검색을 수행하기 위해 카테고리 단위로 설정했다.



How many vectors you need to put into your vector database

[그림 num] 데이터 사이즈 별 Milvus 배포 옵션 추천

## Hybrid Search

검색 모델은 dense와 sparse를 모두 활용하는 hybrid search 방식을 활용했고, 두 score를 weighted sum을 통해 결합했다. 이후 hybrid score 기준으로 top-k search를 진행하여, 장소 후보지를 선정했다.

$$Score = w \cdot Dense\_Score + (1 - w) \cdot Sparse\_Score$$

[그림 num] Hybrid search scoring method

리뷰 검색 시에 겪었던 가장 큰 문제는 긍·부정 의미 반영 문제이다. 검색 모델은 의미적 유사성은 잘 파악하지만, 해당 문장의 긍·부정 여부에 대해서는 판단할 수 없었다. 예를 들어, “맛있다”라는 내용을 검색했을 때 “맛없다”, “맛있지 않다” 등도 검색 대상에 포함되는 것이었다. 따라서 해당 문제를 해결하기 위해, 리뷰 요약을 긍정 요약과 부정 요약으로 나누어 저장하고 각각의 내용과 유사도 계산 후 두 값의 차를 통해 dense score를 계산했다. 이를 통해, 긍정 리뷰 부분과 유사도가 높고, 긍정적인 리뷰가 많은 장소에 더욱 가중치를 부여할 수 있었다.

$$Dense\_Score = \frac{Pos\_Cnt}{Pos\_Cnt + Neg\_Cnt} Pos\_Score - \frac{Neg\_Cnt}{Pos\_Cnt + Neg\_Cnt} Neg\_Score$$

[그림 num] Dense retrieval scoring method

# TMap API를 활용한 후보 장소 선택

초기 아이디어에는 카테고리별로 **Retrieval**로 통해 탐색된 장소 중 가장 첫번째 장소를 선택하는 방향으로 설계했으나, 일반적으로 다음 장소를 선택할 때에는 현재 장소 위치를 기준으로 다음에 갈 장소를 선택하기 때문에 이와 동일하게 구현하였다.

## TMap API

장소 간 거리를 구할 때에는 직선 거리가 아닌 실제 지도의 길찾기 기능과 같이 소요 시간이 필요하다. 실제 도보 거리를 기준으로 후보 장소를 선택하기 위해 **Map API**를 사용하였다. 현재 서비스되고 있는 API는 **Naver, Google, Kakao, Tmap API**가 있었으나, **Naver**와 **kakao**는 이동 수단이 자동차(택시)만 존재하였고, **Google**의 경우 자동차, 도보, 대중교통 모두 존재하였으나 국내에서는 대중교통만 제공되기 때문에 도보 거리를 측정할 수 있는 **Tmap API**를 선택하였다.

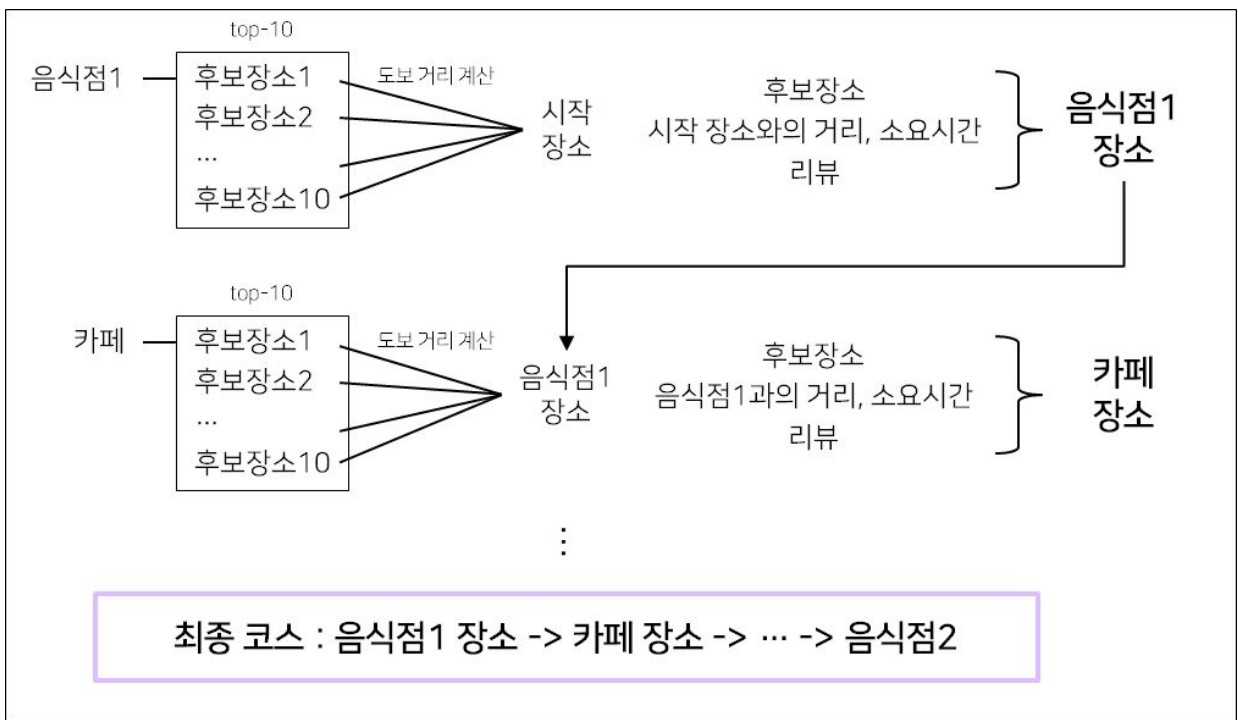
## 장소 선택

이전 카테고리에서 선택된 장소를 기준으로 거리를 계산하여 다음 장소를 선택한다. 처음 시작장소로부터 **Retrieval**로 탐색된 카테고리별 후보 장소간의 거리를 구해 리뷰가 포함된 후보 장소, 시작 장소와의 거리, 소요시간을 **LLM Prompting**을 통해 장소를 선택한다. 선택된 장소는 그 다음 장소를 선택하기 위한 기준 장소가 되며, 생성된 카테고리 끝까지 반복한다.

장소를 선택하는 기준에 대해서는 다음과 같은 시스템 프롬프트를 두었다.

### 추천 원칙

- 요구사항과 장소의 리뷰가 얼마나 잘 맞는지를 가장 중요하게 평가합니다.
  - 장소의 분위기, 시설, 특징 등이 요구사항과 얼마나 일치하는지 확인합니다.
- 유사한 후보가 여러 개라면 더 가까운 곳을 추천합니다.
  - 같은 조건이라면 거리나 이동 시간이 짧은 곳이 더 좋은 선택입니다.
- 추천 이유는 "요구사항 부합"과 "접근성" 두 가지 기준으로 설명합니다.
  - "요구사항 부합": 장소의 특징이 사용자의 요청과 어떻게 맞는지 설명합니다.
  - "접근성": 다른 후보들과 비교했을 때 거리나 소요시간이 왜 적절한지 설명합니다.



[그림 num] 장소 선택 과정

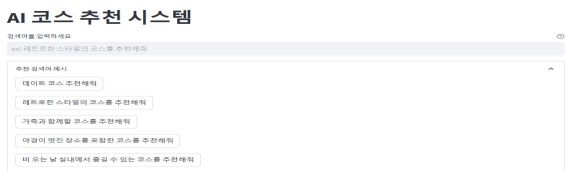
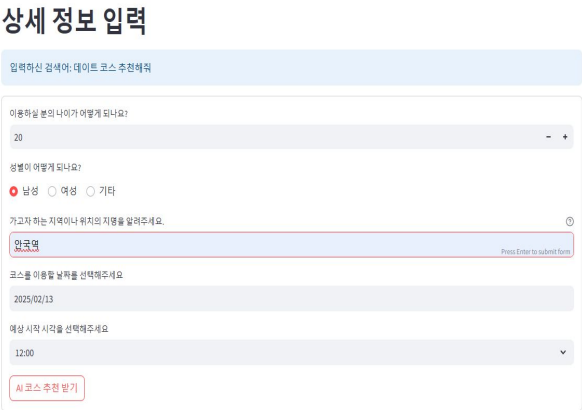
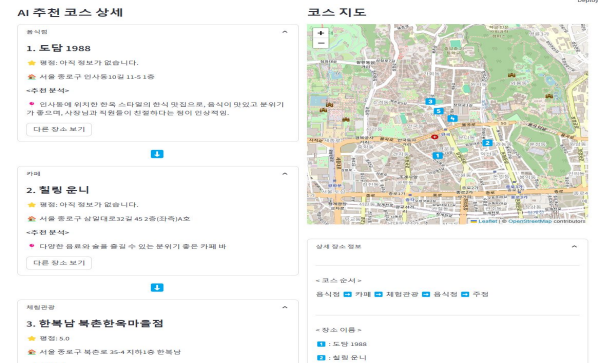


# 데모 UI 구현

## 개요

WITY의 UI는 사용자가 원하는 취향을 AI에게 직관적으로 제공할 수 있도록 구성되었다. 처음 계획시, 주어진 예상 구현 기간이 일주일 인 것을 감안하여 **Stimlit**을 채택한 웹 상의 서비스로 구현되었다. 먼저 어떤 느낌의 코스를 원하는지를 자유롭게 적은 후에 실제 이용자의 기본 사항을 추가로 제공받는다. 해당 내용을 토대로 코스가 완성되면 마지막 UI에서는 코스정보를 파악하기 쉽도록 왼편에는 상세내용, 오른편에는 지도상의 실제 위치 정보를 준다.

## 화면 기능 설명

<UI>	<상세설명>
<div>1, 검색화면 UI</div> <div></div>	<p>검색어 입력 : 사용자는 원하는 코스의 느낌을 자유롭게 입력한다</p> <p>검색어 예시 : 코스를 만들고 싶지만 어떤 질문을 해야할지 막막할 때, 가족, 연인, 친구의 테마를 중심으로 사용자가 고를 수 있는 버튼을 구성하였다.</p>
<div>2, 추가정보 UI</div> <div></div>	<p>실제 코스 이용자 정보 입력 : 연령대, 성별, 가고자하는 지명, 예상 코스 시작 날짜와 시간을 입력 받는다. 해당 정보들은 어머니를 위하여 코스를 생성하는 자녀와 같이 서비스를 이용하는 사용자와 실제 코스를 이용하는 사용자가 다른 경우까지 생각하여 구성하였다. 또한, 각 항목의 정보를 통하여 <b>Safety</b>관련 상황을 코스생성시 피하고자 한다. 다음은 각 정보를 통하여 제한을 둔 예시이다.</p> <ul style="list-style-type: none"><li>연령대 : 10대에게 술집 추천 금지</li><li>성별 : 남성에게 여상들만 이용하는 장소 피하기</li><li>지명 : 혜화역 근처에서 놀고 싶은 상황에서 다른 지역 추천 금지</li><li>예상 날짜와 시작 시간 : 두시이후 시작 시간일 때 점심을 이미 먹은 상황에서 점심식사 장소로 코스의 시작 금지</li></ul>
<div>3. 결과 UI</div> <div></div>	<p>코스 정보(왼쪽): 코스의 각 장소에 대한 기본 정보 및 리뷰기반으로 요약된 추천하는 이유를 제공한다</p> <p>다른 추천 장소(왼쪽): 추천이유나 동선의 문제 등의 이유로 해당 장소가 마음에 들지 않을시 해당 카테고리과 같은 다른 장소들을 추천한다. 장소 후보군 중 마음에 드는 장소를 누르면 기존의 코스관련 테이블에 즉시 반영이 된다.</p> <p>지도 (오른쪽): 코스 순서대로 번호가 표시되며, 하단에 카테고리별 코스 순서와 장소명이 제공된다.</p> <p>기타 버튼 (하단): 다시 검색하기, 상세 정보 수정, 현재 코스 저장 버튼이 있다. 이 중 ‘현재 코스 저장 버튼’을 사용시 한번 저장해 두면 왼쪽에서 수정 후 마음에 들지 않거나 현재 코스와 비교할 수 있는 상황을 가정하여 추가하였다.</p>

# 리뷰 요약 평가

HyperCLOVA X (HCX-003)를 활용하여 원본 리뷰 데이터를 긍정 리뷰 요약, 부정 리뷰 요약, 긍정 리뷰 개수, 부정 리뷰 개수, 전체 리뷰 요약으로 변환한 후, 각 요약의 품질을 평가하였다. 이를 위해 총 리뷰 개수를 기준으로 20개 이하 및 20개 이상 두 개의 그룹으로 데이터를 분류하고, 각 그룹에서 300개씩 샘플링하여 25회 반복 스코어링을 수행한 후 가중평균을 적용하였다.

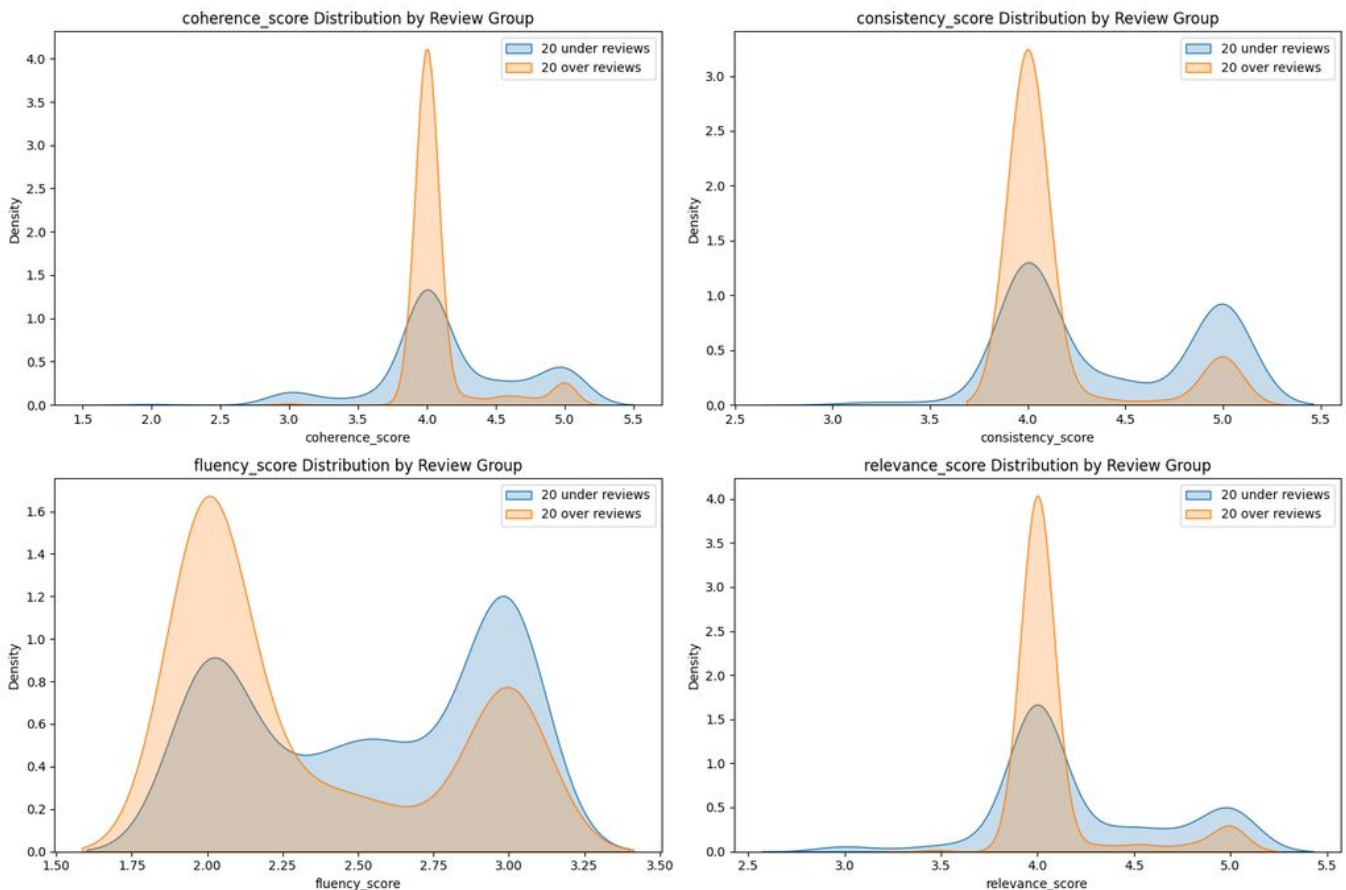
2. 평가 기준 LLM을 이용한 리뷰 요약 평가에서는 다음 네 가지 기준을 적용하였다.

- **Coherence** (일관성, 1~5점): 긍정/부정 요약과 전체 요약이 논리적으로 연결되는가?
- **Consistency** (사실성, 1~5점): 원본 리뷰에 기반한 사실적 일치 여부, 환각 정보 여부
- **Fluency** (유창성, 1~3점): 문법, 맞춤법, 문장 구조 등의 자연스러움
- **Relevance** (관련성, 1~5점): 원문의 핵심 내용을 명확히 반영하는가?

3. 분석 결과 평가 결과, 리뷰 개수가 20개 이하인 데이터에서 높은 스코어가 나타났으며, 이는 특히 일관성(**Coherence**)과 관련성(**Relevance**) 지표에서 더욱 두드러졌다. 반면, 리뷰 개수가 많은 경우 부정적 의견을 요약할 때 세부적인 정보가 누락되거나 중복이 발생하는 경향이 확인되었다.

4. 통계적 검정 결과 독립표본 t-검정을 수행한 결과, 리뷰 개수가 20개 이하인 그룹이 20개 이상인 그룹보다 유의미하게 높은 평균 점수를 보이는 것으로 나타났다. 특히, **\*\*Coherence, Consistency, Relevance** 항목에서 유의미한 차이( $p<0.05$ )가 존재함을 확인하였다.

5. 결론 및 제언 본 연구 결과를 종합하면, 리뷰 개수가 많을수록 요약 품질이 떨어지는 경향이 있으며, 특히 부정적 의견을 요약할 때 내용이 중복되거나 일부가 누락되는 문제가 발생할 수 있음을 확인하였다. 따라서, 적절한 리뷰 개수를 설정하여 요약하는 것이 더 높은 품질의 요약을 생성하는 데 유리할 것으로 보인다. 향후 연구에서는 최적의 리뷰 개수를 정량적으로 결정하는 방법론을 도입할 필요가 있다.



# 시나리오 데이터 생성

## 시나리오 데이터 생성 배경 및 방법

실제 사용자의 요구 사항과 이에 대응하는 경로가 없기 때문에 평가를 위한 데이터가 필요하다. 이에 다양한 상황에 맞게 제대로 된 경로를 추천하는지 평가하기 위해 데이터를 생성하였다. LLM을 이용하여 ‘사용자 요구사항’, ‘성별’, ‘연령대’, ‘일정 시작 시각’을 포함한 시나리오를 생성하였다.

각 ‘성별’, ‘연령대’, ‘일정 시작 시각’은 랜덤하게 생성하도록 하였고, 첫 데이터 생성 시에는 구체적인 ‘사용자 요구사항’을 생성하도록 하였다.

이상적인 시나리오는 사용자가 구체적인 요구사항을 입력하는 것이지만, 실제 사용자는 그러하지 않을 가능성이 높기 때문에 첫 번째 데이터에서 일부 요구사항을 제거하여 단순한 요구사항으로 바꾸는 작업을 하였다.

## 생성된 시나리오 데이터 개요

### Origin Dataset

- 사용자 요구사항  
사용자의 선호, 신체적 제약, 기념일/특별한 이벤트, 감정 상태, 환경 선호 중 1~2개 정도의 요구사항이 들어가도록 생성
- 성별  
‘남성’, ‘여성’, ‘혼성’ 중 ‘사용자 요구사항’과 일치하는 연령대가 되도록 생성
- 연령대  
10대에서 60대 중 ‘사용자 요구사항’과 어울리는 연령대가 되도록 생성
- 일정 시작 시각  
9시 ~ 18시 중 ‘사용자 요구사항’과 어울리는 시간대가 되도록 생성

### Strange Dataset

위 ‘Origin Dataset’의 ‘사용자 요구사항’만 다음 방식으로 다르게 생성하였다.

- 더 짧고 대충 입력한 형태로 변형
- 문장에 다소 어색한 표현 추가
- 감정적인 표현이나 일상적인 말투 추가
- 중요한 필요 정보를 일부 생략
- 애매하거나 부족한 정보가 포함되도록 변형

## 시나리오 데이터 예시

	Origin Dataset	Strange Dataset
사용자 요구사항	기분 전환 겸 혼자서 나들이 하고 싶은데 어디가 좋을까요? 전시 작품도 보거나 커피도 마실 수 있는 곳이 근처에 있으면 좋겠네요.	전시회 볼만한 데 없어? 주변에 카페도 있으면 좋겠어.
성별	여성	
연령대	20대	
일정 시작 시각	13시	

# 카테고리 기반 코스 생성 능력 평가

## Evaluations

### LLM Evaluation

우리 서비스는 하루의 계획을 하나의 경로 형태로 생성해 제시한다. 일반적으로 하루 계획에 대한 경로는 정답을 갖지 않는다. 경로를 하나의 **gold-route**로 제한한다면, 평가의 신뢰성을 떨어뜨리고 경로의 유연성을 훼손시킬 것이다. 이러한 이유로, LLM의 카테고리 경로 생성 능력 평가를 위해 **guide-line** 기반의 LLM 평가를 선정했다. 경로를 만들 때 고려해야 할 기본 사항을 **guide-line**으로 설정하고, 이를 기준으로 LLM 평가를 진행하는 방식이다.

당신은 일정 및 계획을 평가하는 감독관입니다. A-B-C-D와 같은 경로 형태의 일정이 주어졌을 때, 해당 경로가 적절한지에 대해 평가해주세요. 경로에 대한 평가는 주어진 조건과 입력되는 사용자의 연령, 성별 등의 정보, 일정 시작 시간 등을 고려해서 평가해주세요. 만약 경로가 주어진 조건에 부합하고 적당하다고 판단된다면 \$\$\$\$1\$\$\$\$\$으로, 적당하지 않다고 판단되면 \$\$\$\$0\$\$\$\$\$으로 응답해주세요.

###조건###

- 주어진 시간 대비 너무 많은 경로 혹은 적은 경로는 생성되지 않아야 한다.
- 구체적인 요청이 있는 경우를 제외하고, 같은 카테고리를 반복해서 추천하지 않는다.
- 16대 사용자에게 주점을 추천하지 않는다.
- 임신, 높은 연령대, 장애 등으로 인한 행동 제약이 있을 시 활동적인 카테고리를 추천하지 않는다.
- 활동적인 일정이 연속적으로 배치되지 않도록 한다.
- "부모님과 함께", "아이와 함께"와 같은 동행자의 나이를 추측할 수 있는 맥락이 있다면, 그 부분을 함께 고려한다.
- "카페가 좋아" 등 사용자의 특별한 요구사항 및 선호도가 반영되어야 한다.
- 식사, 음주 등은 적절한 시간대에 고려되어야 한다.

###예시###

###예시1###

\*\*\*\*사용자 정보\*\*\*\*

나이: 40대  
성별: 여성  
요구사항: 임신한 아내와 함께 산책하면서 예쁜 풍경도 보고 싶은데 편함은 잘소 있나요?

\*\*\*\*시간 정보\*\*\*\*

일정 시작: 13시  
일정 끝: 22시

\*\*\*\*경로 정보\*\*\*\*

경로: @@채널관광-공연전시-음식점-카페@@

\*\*\*\*평가 결과\*\*\*\*

주어진 경로는 @@채널관광-공연전시-음식점-카페@@입니다. 사용자 요구사항에는 임신한 아내와 동행한다는 내용이 포함되어 있습니다. 하지만, 해당 경로에 포함된 "채널관광", "공연전시"는 임신부에게 지나치게 활동적인 활동이 될 수 있고, 이는 조건 %%%에 부합하지 않습니다. 따라서, 해당 경로는 적당하지 않은 경로이고, 최종 결과는 \$\$\$\$0\$\$\$\$\$입니다.

결과: \$\$\$\$0\$\$\$\$\$

:

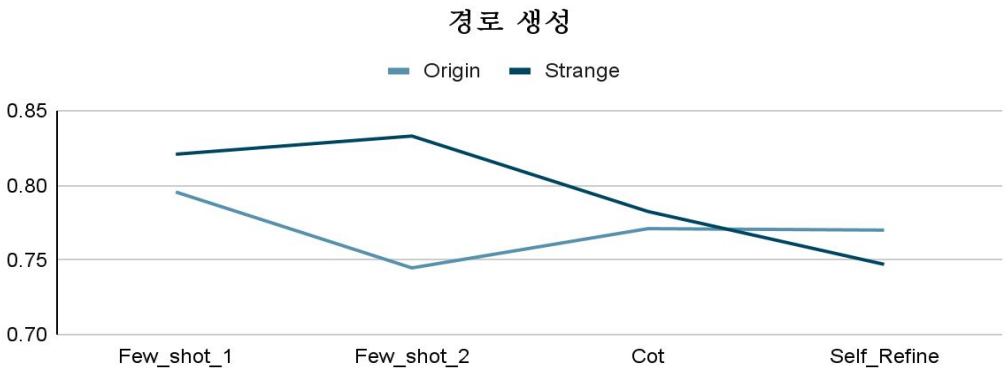
(few shots)

[그림 num] System prompt for evaluation

Parameter	Value
model	gpt-4o
max_tokens	512
temperature	0
top_n	1
frequency_penalty	0
presence_penalty	0
stop	None
logprobs	True
top_logprobs	10
n	1

[그림 num] Parameters for OpenAI API model

다음은, 카테고리 생성 과정의 각 **method**에 대한 평가 결과이다. 먼저, **Origin**의 경우 매우 구체적이고, 디테일한 고객 요구 사항이 주어지는 경우를 상정했다. **Few-shot**과 **CoT**의 경우 80%에 근접하는 성적을 얻을 수 있었다. 다음으로 **Strage**의 경우, 간결한 요구 사항이 주어지는 경우를 상정했다. 요구 사항이 많지 않기 때문에 평가할 사항이 많지 않고, **orgin**과 비교하여 전반적으로 점수가 높은 것을 확인할 수 있다. 추가적으로 두 경우 모두 성능 향상을 위해 **Self-Refine**을 적용시켰으나, 충분한 향상을 이끌어내지 못했다. 추후, **Self-Refine**을 더욱 고도화 하거나 **guide-line**을 평가가 아닌 **Self-Refine** 과정에 포함시켜 더욱 성능을 개선할 수 있을 것으로 기대된다.



[그림 num] Evaluation results for each methods

# 후속 개발 사항

## Reranker

현재 LLM을 내부 지식 및 자유로운 **output** 형식 기반 빠른 개발 목적으로 활용하여 카테고리 경로 생성, 검색된 장소 기반 최종 경로 생성 단계에서 사용하고 있다. 카테고리 생성의 경우에는 LLM이 스스로 판단하여 카테고리 생성하는 것이 필요하지만, 후보 장소 중 추천할 장소를 선택하는 과정은 **Reranker** 학습을 통해서 **Query**에 대한 적합도를 추가 고려하여 해당 과정을 최적화할 수 있을 것이다.

## Retrieval 고도화

리뷰에 대한 긍정, 부정 요약 각각의 유사도와 긍정, 부정 리뷰 개수를 활용한 **weighted sum** 방식으로 현재 긍정, 부정 검색을 수행하고 있다. 하지만, 리뷰와 사용자 요구사항의 **aspect**를 고려하지 않아, 주요한 내용 외에서 유사도 발생이 가능하다. 예를 들어 “가성비 좋은 음식 점을 추천해줘”라는 **Query**가 입력되면, “좋은 음식점 추천” 부분과 유사한 검색이 이루어질 가능성이 높다.

따라서, 전체 리뷰 데이터에 대해 긍정, 부정 평가 후, 각각 사용자 요구 사항과의 유사도 합산이 필요하고, 리뷰와 사용자 요구사항에 대한 **aspect**를 분석해 **aspect**가 일치하는 리뷰에 대해서 검색을 수행하는 방식을 개발하고자 한다.

## Dataset

현재 데이터는 카테고리 구분이 다양하지 않고 모호하며, 특정 카테고리에 대한 데이터가 밀집된 상태이다. 작업 순서 상 카테고리 경로가 먼저 생성되고 실제 장소들이 검색되는데, 카테고리에 해당하는 장소가 없는 문제가 발생하기도 한다. 또한, 지도 서비스 상의 리뷰만 활용하여 데이터를 수집하였기에 다양성이 부족한 면이 있다.

데이터를 더욱 균등하게 구분하는 카테고리 체계를 재수립하고 유연한 카테고리 경로 생성을 위해 카테고리에 대한 세분화를 진행할 계획이다. 추가적으로 블로그 리뷰와 같은 다른 매체에서 데이터를 수집하고 가게 휴무일 정보 등의 세부적인 내용을 활용하여 더 디테일한 서비스를 제공하고자 한다.

## 검색 범위 확장

현재 이동 수단은 도보 이동으로만 한정되었기에 **500m ~ 1km** 이내의 장소만 탐색하고 있다. 이 경우, 지역 탐색이 지나치게 제한적이라는 문제점이 발생하게 된다.

도보를 비롯하여, 버스-지하철 등 대중 교통 및 다양한 교통 수단으로 이동 수단을 확장하고 더 넓은 장소를 검색할 수 있도록 구현하고자 한다. 또한, 이동 시간에 대한 패널티를 부여하여, 적절한 이동 시간을 유지하도록 유도하고자 한다.

## 멀티턴 대화 방식

현재 사용자는 요구사항을 제한적으로 작성하여 전달하게 되어 사용자가 원하는 장소가 출력되지 않는 경우가 있다. 이를 멀티턴 방식으로 구현하여 사용자의 요구사항을 더 자세하게 입력받고 입력받은 요구사항을 **Query Rewriting**을 통해 보다 정확한 탐색이 이루어지도록 하고자 한다.



# 개인회고 - 강경준

## 프로젝트 개인 목표

- 단순히 하나의 기술 혹은 하나의 모델을 만드는 것이 아닌, 여러 기술을 통해 파이프라인을 구성하고 이를 적절히 동작시키는 것을 목표로했다. 하나의 서비스를 위해 기술에 대한 고도화도 중요한 역량이지만, 여러 기술들이 유기적으로 연결되어 동작하는 것 또한 필수적인 경험이라 생각한다.
- 또한, 해당 과정에서 발생하는 현실적 문제를 직면하고 이를 해결하는 것을 목표로 했다. 학습하는 입장에서는 정형화된 문제만을 마주하기 마련이다. 하지만, 내가 배운 기술을 실전에 적용하는 과정에서는 다양한 문제가 발생할 수 있다. 이러한 문제들을 정의하고 해결하는 과정은 기술적인 역량과는 또 다른 역량을 개선시킬 것이다.

## 시도 및 결과

- 경로 생성을 위해 **RAG** 적용 -> 결과가 어땠더라
- 리뷰에 대한 유사도 기반의 검색 시스템을 만들기 위해, **VectorDB**를 구축하고 이를 활용한 검색 모델을 구현했다. **VectorDB** 구축 과정에서 발생하는 다양한 옵션들에 대해 고민하고, 더욱 효율적인 데이터 저장 방식에 대해 고민해볼 수 있었다. 또한 검색 모델의 실질적 능력에 있어서, 긍 부정 의미 반영 문제와 검색 의도 반영 문제 등 여러 문제를 경험하고 이에 대한 해결 방안 고민해볼 수 있었다. 이에 대한 결과로 데이터 저장 효율성을 **50%** 가량 개선할 수 있었고, 긍 부정 의미 반영 문제를 간략하게나마 해결해볼 수 있었다.
- 정답이 정해져 있지 않은 하나의 서비스이기 때문에 그 평가 방식에 대해서도 많은 고민이 있었다. 정답을 하나로 고정하여 이를 활용한 학습 또는 평가했을 때, 편향이 발생하거나 평가 지표의 신뢰성이 떨어지는 문제가 발생할 수 있었다. 이로 인해 **LLM**을 활용한 평가를 고려했고, **G-eval** 등의 접근에 대해서 알게되었다. 또한, 실제로 최고 수준의 **LLM**을 활용해보면서 이에 대한 활용성과 비용에 대한 감각을 기를 수 있었다.

## 좋았던 점과 배운 점

- 아이디어를 도출하고 구체화하는 과정의 중요성을 몸소 체험할 수 있었다. 많은 논의와 고민이 동반되는 작업이었고, 분야나 목적이 특정되지 않는다면 쉽게 결정이 나지 않거나 여러 번 결정을 뒤엎게 되는 일도 있었다.
- **VectorDB**에 대해 기술적 이해도가 높아졌다. 실제 **VectorDB**를 제공하는 시스템의 종류, 선택 해야할 옵션, 제공하는 기능 등에 대한 이해도를 기를 수 있었다.
- 검색 기술 고도화와 관련된 경험을 얻을 수 있었다. 벡터 유사도 기반 검색 시스템이 갖는 문제점을 발견할 수 있었고, 이에 대한 해결방안 까지 고려해볼 수 있었다.

## 아쉬운 점

- 해커톤 기간이 너무 짧다는 점이 가장 아쉬웠다. 아이디어 도출에 소요된 기간을 제외하면, 기술 개발을 위해 사용할 수 있었던 시간이 그리 길지 않았고 결국 많은 것을 도전해볼 수 없었다. 이로 인해, 다양한 기술적 문제들을 추후 개발 과제로 남겨둘 수 밖에 없었다.
- 기술적으로 더욱 고도화될 필요가 있다. 단순히 일반화된 툴을 활용하는 것에 그치는 것이 아니라, 논문 등을 참고해 더욱 발전된 기술을 직접 구현하는 등의 과정을 추가할 필요가 있다.

## 앞으로의 목표

- 검색 모델의 고도화와 관련하여 생각했던 방안을 실제로 적용시켜보는 것이 목표이다. 아무리 구체적으로 해결 방안을 고려한다 해도 실제로 적용시켜 보지 않으면 결과는 알 수 없기 때문에, 이를 직접 적용시켜 구체적인 성능 향상을 이끌어내 보는 것이 목표이다.
- 서비스를 만드는 과정에서 적절한 결과물을 받고 결과물의 퀄리티를 개선하는 일에만 집중했다. 추론 과정의 최적화에 대해서는 많은 고민을 하지 못했기 때문에, 이에 대한 문제 해결 방안을 고민하고 시도하는 것이 목표이다.

# 개인회고 - 권지수

## 프로젝트 개인 목표

이번 프로젝트는 기업 해커톤이었기 때문에 우선 서비스의 방향을 잘 모색하고 완성하는 것을 보는 것이 목표였다. 이번에도 데이터를 담당하고 싶었고, 우리가 진행하기로 한 프로젝트는 리뷰 데이터를 이용하여 코스를 추천하는 서비스여서 데이터를 많이, 그리고 다양하게 수집하고, 중구난방인 데이터를 잘 정제해야겠다고 생각했다. 웹 스크래핑은 여러 사이트에서 해본 경험이 있지만, 이렇게 여러 사이트에서 스크래핑한 데이터를 하나의 데이터로 통합하는 과정은 겪어본 적이 없어서 까다롭겠지만 재밌을 것이라고 생각했다.

## 시도 및 결과

우선 naver, kakao, google에서 6,605개의 음식점 및 리뷰 데이터(30개 썩)를 모두 스크래핑 하고 통합하는 과정이 조금 까다로운 게 아니었다. 각 사이트에서 “음식점+주소”로 검색한 결과를 스크래핑 했는데, 이름이나 주소가 미묘하게 달라 같은 가게를 잘 스크래핑 했는지 확인하는 수작업이 필요했고, 또 카테고리도 너무 세분화되어있어 이 부분도 일일이 확인해서 대분류 컬럼을 추가하고 분류하는 작업을 해야 했는데, 결과적으로 이러한 수작업들에 많은 시간을 들이게 되었다. 수작업이 끝난 뒤에는 장소 정보 데이터 중 영업시간에 대한 내용 형식이 다 달라서 이 부분은 모두 같은 형식으로 통일하는 작업을 진행했다. 리뷰 데이터는 사람이 직접 기분에 따라 서술한 것이라 여러 모양의 이모지 및 이모티콘, 그리고 과다한 특수문자들이 섞여 있어 이를 제거하는 과정을 거쳤다. 우리는 리뷰 데이터로 긍/부정으로 나눈 뒤 이후 태스크를 진행할 예정이어서 최대한 글자만 남기게 되었다. 데이터를 정제하는 부분은 룰베이스(rule-based)로 진행했다. 마지막으로 전체 데이터를 통합할 때에는 리뷰 데이터가 없으면 그에 해당하는 장소 데이터도 삭제 처리하여 완성했고, 이를 최종 데이터로 이용하게 되었다.

## 좋았던 점과 배운 점

이번 프로젝트에서 데이터를 수집하는 정제하는 시간이 너무 길어졌기 때문에 걱정이 많았는데, 웹 스크래핑을 다같이 끝내고 내가 정제를 시작함과 동시에 다른 팀원들은 맡은 태스크를 진행하기 시작했다. 그래서 최소한의 데이터를 가지고 각자의 모듈을 구축한 뒤 후에 최종 데이터로 통합했다. 이렇게 나누어 작업해서 결과적으로 완성할 수 있었고, 무사히 서비스를 모두 구축할 수 있었다고 생각한다. 그리고 서비스를 실제로 만들면서 기술적인 부분 말고도 **Safety**와 같은 부분에서 고려할 점이 많다는 것을 깨달았다. 실제로 유저가 사용했을 때, 서비스가 코스를 유저와 어울리지 않게 생성될 수도 있다는 것을 확인했고, 이를 고쳐나가는 점이 흥미로운 부분 중 하나였다.

## 아쉬운 점

우리의 서비스는 우선 사용자 쿼리에 맞는 코스의 대분류를 구성한 뒤, 그 분류에 맞는 장소를 추천하는 순서로 진행이 되었는데, 하다보니 데이터의 대분류 중 너무 적은 수의 장소가 있는 카테고리는 특정 장소에서 구성을 할 때 데이터가 없어 구성이 안되는 경우가 있었다. 이 부분을 처음에 데이터 카테고리 분류 작업할 때 미리 예상하지 못한 것이 아쉬웠다. 이걸 예상할 수 있었다면 너무 적은 수의 장소가 있는 대분류는 데이터를 더 수집하거나, 카테고리 분류를 다른 방식으로 했을텐데 하는 후회가 조금 남았다. 이 문제는 결국 다른 팀원들의 능력으로 기술적인 부분을 조금 변경해서 해결하게 되었지만, 그래도 데이터 부분을 맡았던 내가 미안했던 부분이었다.

## 앞으로의 목표

해커톤을 마지막으로 네이버 부스트캠프를 마치게 되었다. 그동안 좋은 강의, 그리고 좋은 동료들을 만나 여러 프로젝트를 하며 관련 지식뿐만 아니라 협업툴, 협업 방법에 대해 알게 된 것 같아서 뜻깊은 시간이었다. 이 곳에서 배운 모든 것들을 이력서에 잘 녹여내서 취업 전선에 뛰어들 예정이다. 앞으로 또 이런 기회를 얻기 힘든만큼 포트폴리오를 잘 작성해 보려고 한다. 취업 전까지는 추가적으로 자격증과 어학 성적을 위해 공부할 예정이다.

# 개인회고 - 김재경

## 프로젝트 개인 목표

이번 프로젝트에서는 일상 속 불편함을 해결하는 AI 기반 서비스 개발을 목표로 하였다. 단순한 모델 개발이 아니라, 기획부터 데이터 수집, 모델 학습 및 평가, 최종 서비스화까지 **End-to-End** 경험을 쌓는 것이 핵심이었다. 특히, **\*\*네이버 클라우드 플랫폼(NCP)\*\***을 활용하여 리뷰 요약 및 평가 시스템을 구축하고, 모델의 성능을 통계적으로 검증하는 과정까지 경험하고자 했다.

## 시도 및 결과

### 1. 네이버 클라우드 플랫폼 활용

- HyperCLOVA X를 사용하여 리뷰 요약 및 평가 시스템을 개발하였다.
- 이미지 분위기 추출 기능도 시도했으나, 모델 성능 및 일정 이슈로 제외되었다.

### 2. 리뷰 요약 및 평가 시스템 구축

- 리뷰 개수를 기준으로 20개 이하, 20개 이상 두 그룹(각 300개)으로 나누어 평가를 진행하였다.
- 25회 반복 평가 후 가중 평균을 계산하여 요약 품질을 분석하였다.
- 결과적으로 리뷰 개수가 적은 데이터의 평가 점수가 더 높음을 확인하였다.

## 좋았던 점과 배운 점

1. 기획부터 서비스화까지의 전체 과정 경험
  - 단순한 모델 개발이 아닌, 실제 서비스 관점에서 고민하는 과정을 경험할 수 있었다.
2. 네이버 클라우드 플랫폼 활용
  - HyperCLOVA X를 통해 대규모 언어 모델을 실무에 적용하는 과정을 배웠다.
3. 데이터 기반 의사결정 경험
  - 평가 지표 부재로 인해 G-Eval 및 통계적 가설 검정을 활용한 데이터 분석을 수행하였다.

## 아쉬운 점

### NCP API 호출 지연

- 한 계정을 여러 명이 사용하며 API 호출 딜레이가 발생하였다.

### VLM 미지원

- 이미지 분석 기능을 최종적으로 적용하지 못한 점이 아쉬웠다.

### 모델 성능 한계

- 부정 리뷰 요약에서 정보 누락 및 중복 문제가 발생하였다.

### 데이터 부족으로 인한 추천 시스템 문제

- 특정 장소가 반복 추천되는 문제가 발생하여 개선이 필요하다.

## 앞으로의 목표

부스트 캠프 수료 이후 나는 우선 학교에 복학할 예정이다. 이후 졸업하여 대학원 진학예정이다. 대학원을 가야할지 취직을 해야할지 갈피를 못잡던 중 부스트캠프에 참여하면서 많은 경험을 쌓고, 좋은 사람들을 만나면서 정말 뜻깊은 경험을 하게 되어서 성공적인 6개월을 보냈다고 자부한다.

# 개인회고 - 박동혁

## < ‘UI’의 신비로움 경험 >

UI를 맡으면서 가장 흥미롭고 동시에 당황스러웠던 점은 웹 페이지 간, 그리고 UI와 LLM 결과 간의 데이터 공유 방식이었다. 기존에 안드로이드 앱을 간단히 만들어본 경험이 있어, **Streamlit**을 사용해 각 페이지별로 개별 .py 파일을 만들어 진행했다. 하지만 사용자가 추가 정보를 입력하고 버튼을 눌러 **RAG**로 데이터를 전달하는 과정에서, 버튼이 정상적으로 작동하지 않고 첫 번째 페이지로 돌아간 후에야 실행되는 문제가 발생했다.

이 문제를 해결하기 위해 구글링과 **ChatGPT** 검색을 시도했지만, 명확한 해결책을 찾기 어려웠다. 그래서 처음 실행부터 웹 페이지 간 이동 로직과 데이터 흐름을 로그로 분석하고, 전체 구조를 그림으로 시각화했다. 그 결과, 모든 페이지를 개별 파일로 나누는 방식이 문제가 되었음을 알게 되었다. 특히, 처음 페이지가 아닌 중간 페이지에서 **RAG**로 데이터를 보내면서 예상치 못한 동작이 발생했던 것이다.

**Streamlit**에서는 기본적으로 처음 실행된 페이지를 통해 다른 시스템과 데이터를 공유할 수 있다. 데이터베이스(DB)를 구축해 흐름을 조절하는 방법도 고려했지만, 프로젝트 규모를 감안했을 때 과하다고 판단해 배제했다. 대신, 팀원의 조언을 받아 모든 페이지를 하나의 파일에서 관리하는 방식으로 변경했고, 이를 통해 **RAG**와의 데이터 공유 문제를 해결할 수 있었다.

또한, 프로젝트 일정상 여러 파트를 동시에 구현해야 했기에, **RAG**의 결과물을 예상하여 UI를 먼저 구현하는 방식도 경험했다. 하지만 실제 테스트 과정에서 예상과 달리, **RAG**를 통해 전달된 딕셔너리 키(**type키**)의 값 중 하나가 **string**이 아닌 **list**로 반환되면서 데모가 정상적으로 작동하지 않는 문제가 발생했다. 데이터 형태를 잘못 예측한 탓에, 결국 데이터 전달 구조를 수정해야만 했다. 이 과정에서 입력 데이터의 형태를 단순히 가정하고 코드를 작성하는 것과, 정확한 예제를 기반으로 구조를 명확히 이해한 후 구현하는 것의 효율 차이를 몸소 경험할 수 있었다.

## <아쉬움 : LLM Generation 평가에 대한 도전>

LLM에 대하여 평가를 한다는 것은 비록 공식적인 결과물에는 들어가지 않았지만 개인적으로 “AI서비스가 개척되고 있는 것 만큼 평가 또한 아직 풀리지 않은 미지의 **task**”라는 것을 알게 해준 뜻깊었던 시도들이었던 것 같다.

생성형 AI를 평가하는 방법들은 **Geval** 부터 시작하여 정량적 지표, 리더보드 까지 다양하게 존재한다. 하지만 현재 AI서비스를 평가 하는 방식에서 비교 데이터나 수치가 존재하지 않으면 수치적 평가 및 리더보드를 만들기 까지 시간과 비용이 많이 들어 결국 LLM을 통한 평가가 유일하다.

여기서 LLM을 LLM을 평가하는데 있어서 그 평가에 대한 정당성은 과연 어떻게 획득을 해야하는가에 대한 흥미로움이 생겨 전체 서비스, 리트리버, 리뷰요약 결과에 대한 평가 방법 실험에 도전하게 되었다.

특히 3가지 중 제일 흥미로웠던 “AI 서비스 전체”를 평가해보는 방법론은 우리 팀과 같이 작은 규모이고 비교점이 애매한 서비스인 상황에서 **AB** 테스트가 힘들다는 것을 중심문제로 설정하여 시도하였다. 먼저 우리 서비스의 특징을 기반으로 LLM(**ChatGPT**, **Gemini**) 모델로 페르소나들을 만든다. 이후에, 각각의 가상 인물들이 우리 서비스를 실제 이용하는 가상의 시뮬레이션 상황을 **CoT** 방식으로 설정하여 질문을 하게 한 후, 결과를 다시 해당 LLM에 넣어 해당 가상인물이 얼마나 결과에 대한 만족도 점수와 그렇게 생각한 이유(개선점, 실제 이용 가능성, 불편한점)를 얻어내어 가상의 테스트를 진행하는 것이다. 이를 통하여 시스템의 보완점을 찾아보려는 목적이었다.

5명의 가상의 인물을 설정하여 진행한 결과 피드백까지 깔끔하게 했으나 프로젝트 시간관계상 시간이 얼마 남지 않아 더 다양한 케이스를 생성하여 완전히 적용해보지 못한 아쉬움이 있다. 아직 과정에서의 의문점들이 남은 만큼 이는 꼭 다시 도전해 보고자 한다.

# 개인회고 - 이인설

## 프로젝트 개인 목표

이전까지의 프로젝트에서는 주어진 목적에 대해 연구 및 실험을 바탕으로 진행했다면, 이번 프로젝트에서는 실제로 AI를 이용해 해결할 수 있는 문제를 정의하고 이를 실현시켜 직접 서비스화 해보고 싶었습니다. ‘문장 유사도 측정’, ‘Question Answering’ 등과 같은 단순한 문제가 아닌, 직접 A-Z까지 기획하고 그 안에서 생기는 문제들을 해결하며 전체 과정을 만들고 싶었습니다.

## 시도 및 결과

이번 프로젝트에서는 크게 데이터 수집과 생성, 카테고리 기반 경로 생성 태스크를 맡았습니다.

### 데이터 수집

아이디어를 구체화 하는 과정에서 어떤 데이터가 필요한지 판단하고 직접 수집해보는 과정을 겪었습니다. 이전까지는 주어진 데이터를 이용하거나 정제된 데이터를 사용했다면, 이번에는 직접 수집해보면서 데이터의 부재를 어떻게 해결해 나가야 하는지를 배우게 되었습니다. 처음부터 기존 수집 대상을 얻기 보다는 이를 위해 추가적으로 어떤 정보들이 필요하고 그 정보들을 어떤 데이터에서 얻을 수 있는지 생각하면서 더 좋은 데이터를 만들기 위해 노력하였습니다.

### 카테고리 기반 경로 생성

처음에는 사용자 요구사항에 맞는 장소들을 이용해 경로를 생성하는 과정에서 매우 많은 고려 사항이 필요하다는 것을 깨달았습니다. 항상 어려운 문제를 단순한 방법으로 해결할 수 있는지 고민하는 자세를 가지고 있어, 이번에도 카테고리 기반으로 경로를 생성하는 방법을 떠올려 해결하게 되었습니다. 여러 프롬프팅 기법들을 실험해보며 LLM을 이용해 카테고리 기반으로 경로를 생성하는 기능을 개발하였습니다. 다만, 아쉬운 점은 블랙박스 모델을 사용하다보니 고도화된 방법들을 사용할 수 없었다는 것과 데이터 부족 등의 문제로 다소 간단할 수 있는 프롬프팅만을 이용했다는 것입니다. 추후 블로그 게시글을 사용해 RAG로 구현 혹은 이와 관련된 데이터를 생성해 모델을 학습시키는 방향으로 발전시키고 싶습니다.

### 데이터 생성 및 평가

이번 프로젝트에서 제일 문제가 되었던 것은 데이터 부족으로, 이로 인해 평가에도 어려운 점이 있었습니다. 어떤 부분을 평가할 것인가, 어떤 성능이 중요한 것인가에 대해 고민하고 이를 토대로 평가 방식과 평가용 데이터를 생성하였습니다. 각 성능 평가를 위해 필요한 데이터 유형을 판단하고 이를 LLM을 이용해 생성하였습니다. 다만 이번 프로젝트에서는 LLM을 이용해 성능 평가를 진행하였지만, 생각에만 그쳤던 여러 사람을 통한 라벨링과 규칙 기반의 평가 방식을 실현해 성능을 평가해보고 싶습니다.

## 아쉬운 점과 앞으로의 목표

이번 프로젝트에서는 HyperCLOVA X만을 이용해야하는 제약 조건이 있었기 때문에 다양한 방법들을 진행해보지 못한 점이 아쉽습니다. 또한 데이터의 부족으로 인해 대부분의 과정에서 제약 사항과 부족한 점이 많았다는 것도 아쉽습니다. 하지만 이런 상황에서도 최대한 기능이 구현되도록 팀원 모두 노력한 점에서 좋았던 것 같습니다.

저는 대부분 프로젝트에서 리더 역할을 맡아왔지만, 이번에 팀원으로서 따라가는 입장이 되었습니다. 이 과정에서 내 생각을 어떻게 전달하고 효율적으로 소통해야 하는지 많이 고민하게 되었습니다. 또한 같은 이야기를 하더라도 서로 생각하는 방식이 다름을 더욱더 느끼게 되었고, 항상 같은 생각을 가지고 있는지 확인하는 것이 중요하다는 것을 느끼게 되었습니다.

이번 프로젝트를 통해 기술적인 역량뿐만 아니라 협업에서의 자세와 의사소통의 중요성에 대해 더욱 깊이 고민할 수 있었습니다. 팀 내 각각의 역할을 경험하면서 각자의 위치에서 마주하는 고민을 이해하게 되었고, 이를 바탕으로 어떤 방식으로 소통해야 하는지에 대해 많은 것을 배웠습니다. 앞으로 다양한 환경에서 협업할 기회가 많을 텐데, 어느 위치에서든 효과적으로 조율하고 소통할 수 있는 역량을 길러나가고 싶습니다. 팀의 원활한 협업을 위해 상대방의 입장을 고려하며, 유연하고 주도적인 자세를 갖추는 것이 목표입니다.



## 개인회고 - 이정휘

### 이번 프로젝트에서의 목표는 무엇인가?

이전 ODQA나 수능문제 풀이모델 프로젝트에서 RAG에 대해 처음 접하다보니 제대로된 구현을 하기가 어려웠기에 이번에 고도화된 RAG기반 서비스를 만들고자 하였습니다. 또한, PM으로써 jira-confluence를 통해 task 및 코드 형상 관리와 문서 기록에 좀 더 집중하고 역할 분배가 잘될 수 있도록 조율하는 것을 목표로 하였습니다. 그리고 일정이 된다면 Airflow를 사용한 자동화도 해보고 싶었습니다.

### 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

코스 추천 시스템이라는 것 자체가 생각보다 고려해야될 사항이 많았습니다. 어떻게 코스를 생성할 것이며 어떤 방식으로 장소 간 거리를 조율할 것이며, 어떤 것이 가장 최적화된 경로이며, 또한 사용자의 요구사항에 정말 맞는 리뷰가 없다면 어떻게 할 것인가 등등.. 짧은 기간과 적은 데이터로 시도하기에는 어려웠던 태스크였다고 생각합니다. 그리고 무엇보다 데이터 문제가 가장 컸던 것 같습니다. 스크래핑으로 데이터를 수집하다보니 멀티 쓰레드로도 6600개에 대한 장소와 리뷰 데이터를 수집하는 시간이 오래걸렸습니다. 그리고 스크래핑 데이터이다보니 품질 또한 좋지 않았습니다. 만약에 6600개의 많은 데이터가 아닌 카테고리를 균등하게, 좋은 품질의 데이터만 수집했다면 어땠을까? 라는 아쉬움이 있습니다.

그리고 LLM Prompting 위주의 개발이 아쉬웠습니다. 요약이나 다른 부분을 학습을 통해서 만들어내었다면 좀 더 기술적으로 많은 시도를 해볼 수 있지 않았을까 싶습니다.

### 한계/교훈을 바탕으로 다음 프로젝트에서 시도해보고 싶은 점은 무엇인가?

이번 프로젝트에서 시도해보지 못한 AI Agent를 시도해보고자합니다. 실제로 개발한 서비스가 단계별로 구축되어 있는데, 이 단계가 모두가 필요하지 않을 것 같다고도 생각합니다. 예를 들어 특정 카테고리만을 검색한다거나, 여러 코스를 추천해달라거나 그런 경우 LLM 스스로 판단해서 각 기능을 불러와 처리할 수 있다고 생각하고 좀 더 신기술을 사용한다는 것에 메리트가 있다고 생각합니다.

그리고 데이터에 대한 고려를 더 많이 해봐야겠다는 생각을 합니다. 처음부터 6600개가 아닌 10개, 50개, 100개.. 이런식으로 데이터를 계속 확인하면서 프로젝트 계획을 구체화하는 과정이 필요하다고 생각했습니다. 작은 문제로부터 시작하는 습관을 들이려고 합니다.

### 협업과정에서 잘된 점 / 아쉬웠던 점은 어떤 점이 있는가?

Jira와 confluence, Github을 이용해서 지금 현재 남은 작업이 어떤 것이 있는지, 언제까지 작업을 마무리해야하는지, 내가 어떤 작업을 더 맡아서 해야하는지에 대해 확인할 수 있었고 각자 작업이 끝나자마자 다른 작업이 들어갈 수 있도록 했던 것이 좋았습니다. 하지만, 그만큼 기록을 잘해야 병합하는 과정에서 문제가 없는데, 기록이 자세하지 못해서 많이 아쉬웠습니다. 어떤 작업을 할 때마다 상세하게 기록하는 습관이 필요했다고 생각합니다. 그리고 이번 프로젝트에서도 gitflow를 적용해서 브랜치를 관리하였는데, 많은 작업이 한꺼번에 이루어지다보니 제대로 관리가 되지 않았습니다. 빠르게 baseline을 만들고 그 부분에 대해서 하나하나씩 합쳐나가는 과정이 필요했는데, 이에 대해 제대로 하지 못했던 점이 아쉬웠습니다.