

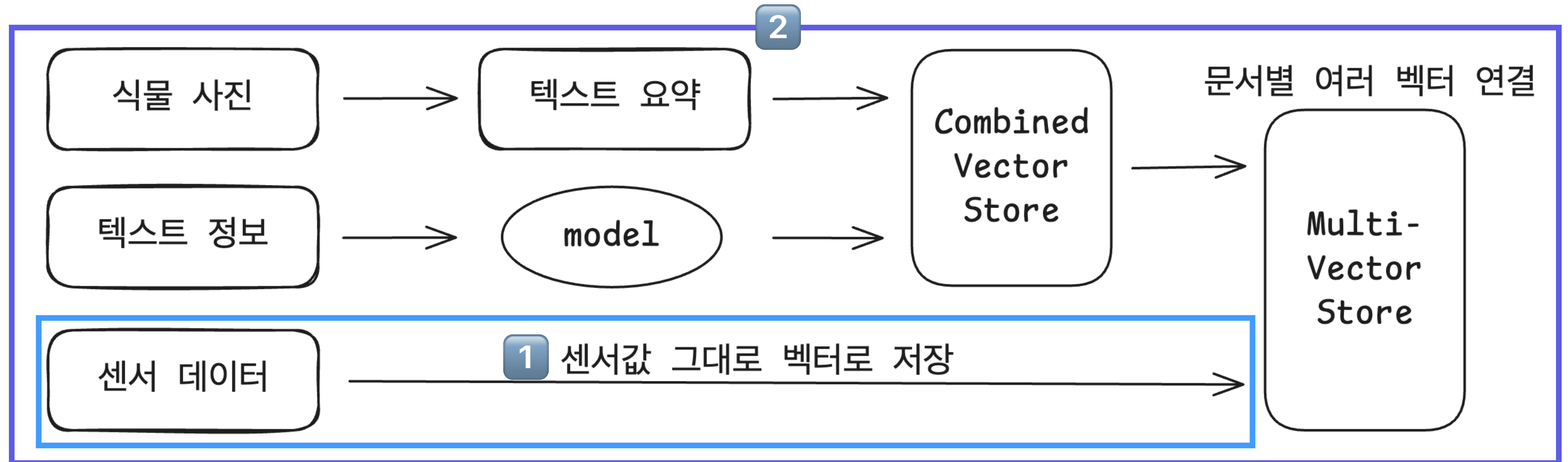
# 2025-2 졸업프로젝트

멀티모달 *RAG*와 성과 기반 분석을 활용한 식물 성장 분석 연구

2025.10.29.

2021104417 우다현

# 이미지 캡션 생성 및 combined vector 저장



# 1 센서값 저장

- 사용한 센서 필드

```
sensor_fields = [  
    "AirTemperature",    # 기온  
    "AirHumidity",       # 습도  
    "Co2",               # CO2 농도  
    "Quantum",           # 일사량  
    "HighSoilTemp",      # 화분 흙 상단 온도  
    "HighSoilHumi",      # 화분 흙 상단 습도  
    "LowSoilTemp",       # 화분 흙 하단 온도  
    "LowSoilHumi"        # 화분 흙 하단 습도  
]
```

## 2 멀티 벡터 저장

```
fields = [  
    FieldSchema(name="id", dtype=DataType.INT64, is_primary=True, auto_id=True),  
    FieldSchema(name="text_vector", dtype=DataType.FLOAT_VECTOR, dim=1024),  
    FieldSchema(name="sensor_vector", dtype=DataType.FLOAT_VECTOR, dim=8),  
    FieldSchema(name="plant_name", dtype=DataType.VARCHAR, max_length=50),  
    FieldSchema(name="growth_level", dtype=DataType.VARCHAR, max_length=10),  
    FieldSchema(name="place", dtype=DataType.VARCHAR, max_length=20),  
    FieldSchema(name="image_name", dtype=DataType.VARCHAR, max_length=100),  
]
```

# 2 멀티 벡터 저장

## plant\_multi\_vector

Collection ID: 461645815217364121



### Status

Ready to load

### Description

Plant multi-vector collection (text + sensor)

### Field Count

7

### Replica

...

### Shards Number

1

### Approx Count

1,429

Field	Type	Nullable	Default	Indexes
id PK	Int64	×	--	Scalar Index
text_vector	FloatVector(1024)	×	--	text_vector IVF_FLAT(COSINE)
sensor_vector	FloatVector(8)	×	--	sensor_vector FLAT(L2)
plant_name	VarChar(50)	×	--	Scalar Index
growth_level	VarChar(10)	×	--	Scalar Index
place	VarChar(20)	×	--	Scalar Index
image_name	VarChar(100)	×	--	Scalar Index

## 2 멀티 벡터 저장

필드	차원	주 사용 목적	인덱스 타입	거리 측정
<code>text_vector</code>	1024	주요 검색 기준 (유사도 검색)	<code>IVF_FLAT</code>	<code>COSINE</code> (텍스트 유사성)
<code>sensor_vector</code>	8	데이터 제공 (필터 후 추출)	<code>FLAT</code>	<code>L2</code> (수치적 거리)

### 인덱스 타입

#### • IVF\_FLAT

- 전체 벡터 공간을 nlist(128)개의 클러스터로 나누어 검색 속도 향상
- 쿼리 시 해당 벡터가 속할 가능성이 높은 클러스터만 탐색
- → 100만 개 데이터 중 일부(≈1%)만 비교, 속도 10~50배 향상

#### • FLAT (for sensor\_vector)

- 차원이 8로 매우 낮기 때문에 클러스터링보다 전체 비교가 더 효율적
- → 정확도 손실 없이 빠른 근거 추출 가능



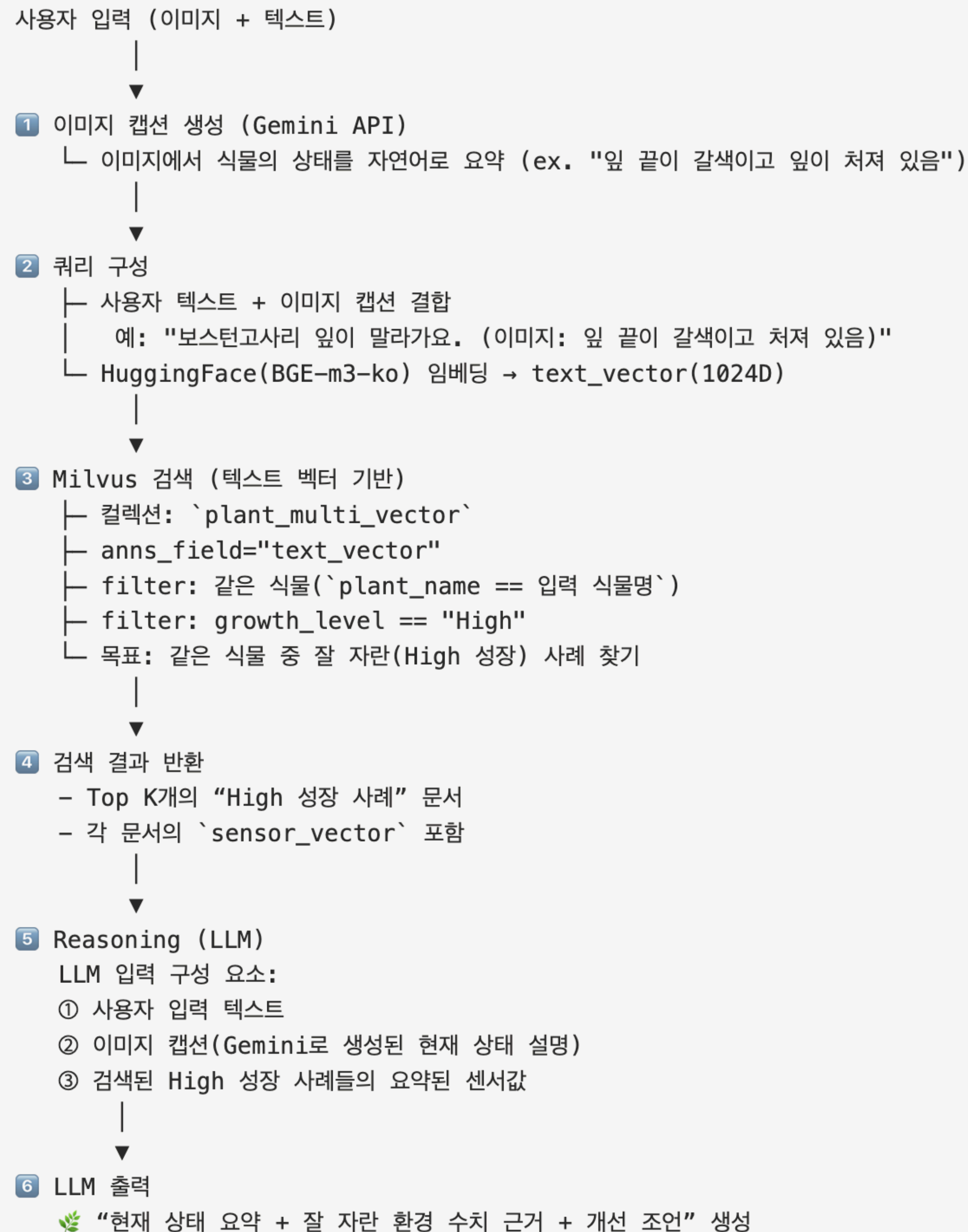
# 3 Retrieval 접근 방식

## 목표:

사용자의 입력(식물명, 상태, 텍스트/이미지)을 기반으로  
같은 식물 중 사용자보다 더 잘 자란(**growth\_level이 높은**) 사례를 찾아,  
그 사례의 환경(센서값)을 “추천 근거”로 제시하는 검색 전략.

## 검색 전략

- ① 사용자와 같은 식물이지만, 성장 수준이 더 높은 사례만 후보로 선정.
- ② 텍스트 기반 의미 검색
  - 사용자 입력(텍스트 + 이미지 캡션)을 BGE-m3-ko 임베딩으로 변환
  - Milvus text\_vector와 cosine similarity 기반 검색.
- ③ 결과 수집 + 센서 데이터 추출
  - 검색된 상위 K개의 High 성장 사례에서 sensor\_vector 추출



## 4 검색 정답 라벨 생성

검색 -> 사용자의 상태와 유사하지만 **더 잘 자란 사례**를 찾아 “비교 근거”로 제시

진단 -> 현재 상태가 좋은지/나쁜지/보통인지 판단하고, 개선 방향 제안

- 검색 정답 라벨 생성 방법 : 정답인 Positive와 오답인 Negative로 구분

```
{
  "query_id": "N50-A-3-08-L-2-V-230828-000366",
  "plant_name": "보스톤고사리",
  "query_growth": "Low",
  "positive": [
    "N50-A-3-08-L-2-V-230917-000906.json",
    "N50-A-3-08-L-2-H-230915-000468.json",
    "N50-A-3-08-L-2-V-230914-000720.json",
    "N50-A-3-08-L-2-H-230915-000487.json",
```



## 5 정확도 측정

검색을 통해 데이터 셋 확보 후

->

10개의 데이터로 시범 테스트 진행

```
=====
🧩 쿼리 텍스트: 보스턴고사리 잎이 시들고 색이 연해졌어요
🌱 식물 이름: 보스턴고사리
📈 현재 성장 단계: Low
🔍 검색 필터: plant_name == '보스턴고사리' and (growth_level
=====
```

🔍 검색 결과 (Top-10):

1. 보스턴고사리 [High] @ 거실(L)	sim=0.323	N50-A-3-08
2. 보스턴고사리 [High] @ 거실(L)	sim=0.343	N50-A-3-08
3. 보스턴고사리 [High] @ 거실(L)	sim=0.344	N50-A-3-08
4. 보스턴고사리 [High] @ 거실(L)	sim=0.344	N50-A-3-08
5. 보스턴고사리 [High] @ 거실(L)	sim=0.347	N50-A-3-08
6. 보스턴고사리 [High] @ 거실(L)	sim=0.347	N50-A-3-08
7. 보스턴고사리 [High] @ 거실(L)	sim=0.348	N50-A-3-08
8. 보스턴고사리 [High] @ 거실(L)	sim=0.349	N50-A-3-08
9. 보스턴고사리 [High] @ 거실(L)	sim=0.349	N50-A-3-08
10. 보스턴고사리 [High] @ 거실(L)	sim=0.349	N50-A-3-08

```
✅ 평가 완료 (10개 쿼리)
{
  "precision@k": 0.7,
  "recall@k": 0.009572649572649574,
  "ndcg@k": 0.7,
  "mrr": 0.7
}
```

# 앞으로의 계획

- 인덱스 방법을 변경하는 등의 유사도를 올릴 수 있는 방법에 대해서 고민
- 프롬프트 작성하기