

# 건설공사 사고 예방 및 대응책 생성

## (한솔데코 시즌3 AI 경진대회)

TEAM : 옥계동

### Content

*Project  
Concept*

*Data  
&  
Technology Stack*

*Model Serving Process  
&  
Architecture*

*Algorithm*

# *Project Concept*

배경

건설공사 사고 상황 데이터를 바탕으로 사고 원인을 분석하고 재발방지 대책을 포함한 대응책을 자동으로 생성하는 AI 모델을 개발

- ✓ 한솔데코는 지속적인 혁신과 기술 개발을 통해 건축 자재 및 안전 관리 분야에서 선도적인 위치를 더욱 공고히 하고자 합니다.
- ✓ 인공지능(AI) 기술을 접목하여 건설 현장의 안전성을 강화하고, 사고 예방 및 대응 체계를 고도화하는 데 주력하고 있습니다.
- ✓ AI의 활용은 건설 현장에서 발생하는 사고 사례를 체계적으로 분석하고, 효과적인 대응책을 자동으로 생성하여 보다 안전한 작업 환경을 조성하는 데 기여할 것입니다.

핵심 기술  
&  
주요 사항

RAG(Retrieval-Augmented Generation) 기반 AI 기술

- ✓ 실제 사고 데이터를 학습하여 건설 현장의 사고 대응 역량을 높이고, 안전관리 프로세스를 고도화
- ✓ 사고 사례 분석과 대형 언어 모델(LLM)을 결합해 보다 정교한 안전 대책 제시

LLM 활용 및 파인 튜닝

- ✓ 안전지침 PDF 데이터를 바탕으로 DAPT(Domain-Adaptive Pre-Training) 방식 적용
- ✓ 안전 지침 사례 데이터를 추가하여 풀 파인 튜닝 수행
- ✓ 전문 분야에 특화된 모델 성능 확보로 정확하고 실용적인 답변 생성

Retriever Model Selection & Tuning

- ✓ 다양한 모델을 시도하여 최적의 Retriever를 선정
- ✓ 앙상블 기법과 가중치 조절 등을 통해 검색 성능을 극대화
- ✓ 현장 특성에 맞춰 커스텀 Retriever를 개발하여 유연한 검색 기능 제공

Post Processing Model

- ✓ 모델의 출력 결과를 사용자 친화적인 형식으로 후처리
- ✓ 현장 작업자와 관리자 모두 쉽게 이해하고 활용할 수 있는 답변 생성
- ✓ 효율적인 커뮤니케이션을 지원하여 안전관리 업무 효율성 향상

## 평가 지표

### 1. S-Bert Cosine 유사도 (CosineSim)

- 예측(생성) 문장과 정답(GT) 문장의 의미론적 유사성을 측정
- 정답 문장의 임베딩 벡터는 jhgan/ko-sbert-sts 모델을 통해 추출
- Embedding Vector는 반드시 Jaccard 유사도 측정을 위한 텍스트와 동일해야 함

### 2. Jaccard 유사도 (JaccardSim)

- 예측(생성) 문장과 정답(GT) 문장에서 사용된 단어 집합의 교집합을 기반으로 어휘적 유사성을 측정

### 3. 최종 평가 점수 (Score)

$$\text{Score} = (1/N) * \sum_{i=1}^N [0.7 \times \max(\text{CosineSim}_i, 0) + 0.3 \times \max(\text{JaccardSim}_i, 0)]$$

## 제약 사항

### 1. 사전 학습 모델 및 API 사용 범위

- 오픈소스 라이선스로 제공되고 가중치가 공개된 사전 학습 모델만 사용 가능
- 원격 서버를 통해서만 접근 가능한 API 형태의 모델(예: OpenAI API, Gemini API 등)은 사용 불가능

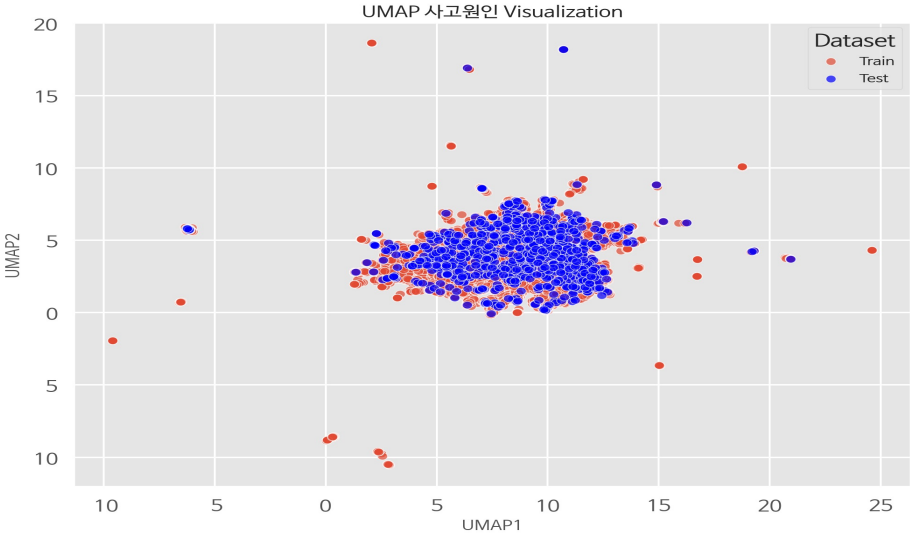
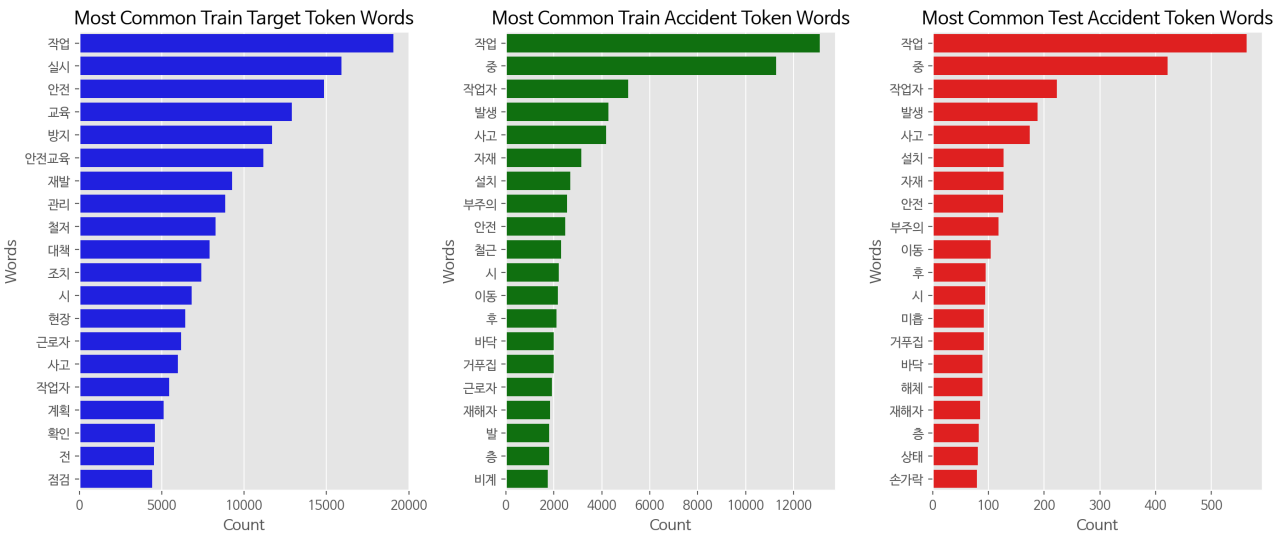
### 2. 외부 데이터 사용 금지

- 대회 제공 데이터만 사용 가능
- 다만, 제공 학습 데이터를 바탕으로 데이터 증강·생성은 가능(단, 사전 학습 모델/API 사용 범위를 준수해야 함)

# *Data & Technology Stack*

## Dataset Info.

- Train Test\_Data 주요 컬럼: ID, 발생일시, 사고인지 시간, 날씨, 기온, 습도, 공사종류, 연면적, 층 정보, 인적사고, 물적사고,공중, 사고객체, 작업프로세스, 장소, 부위, 사고원인, 재발방지대책 및 향후조치계획(Target)
- 추가적으로 train data에서 200개의 sample를 random 추출해 validation data로 활용
- 건설안전지침 : 104개의 건설안전지침 PDF



- 재발방지대책 및 향후조치계획(Target) 텍스트에서는 “작업”, “실시”, “안전”, “교육” 등의 단어가 상대적으로 높은 비중을 차지하는 것으로 확인되었습니다. 또한 사고원인 텍스트의 주요 단어들은 전반적으로 유사한 분포를 보이며, 임베딩 후 차원 축소(UMAP)를 통해 시각화한 결과, Train과 Test 데이터가 유사한 분포를 형성하고 있음을 확인했습니다. 이를 종합해볼 때, 사고원인을 유사도 기반으로 Retriever 방식을 활용하면 보다 효과적인 결과를 기대할 수 있을 것으로 판단됩니다.
- 건설안전지침 데이터는 데이터 형태와 청크 단위 분할, 그리고 Test 데이터와의 유사도 기반 활용에 여러 제약이 있어 현재로서는 Retriever 방식으로 접근하기가 어렵다고 판단했습니다. 다만 추후 모델이 건설안전지침 관련 도메인 지식을 추가 학습할 때에는 유용하게 활용할 수 있을 것으로 기대됩니다.

## Environment



- A100 GPU 사용
- GPU 메모리 40GB

## 데이터 처리 & 분석

- pandas: 데이터프레임 구조로 데이터 처리 및 조작
- numpy: 수치 연산 및 배열 연산
- tqdm: 반복 작업 진행 상황 시각화(Progress Bar)

## 기타

- sklearn.model\_selection (train\_test\_split): 데이터셋 분할
- random: 난수 생성 및 시드 고정
- matplotlib: 그래프 및 시각화

## 자연어 처리 (NLP) & 언어 모델 (LLM)

- transformers (pipeline, AutoModelForCausalLM, AutoTokenizer, BitsAndBytesConfig 등): 다양한 사전 학습 언어 모델 및 토큰라이저 제공
- sentence\_transformers (SentenceTransformer): 문장 단위 임베딩 추출
- bitsandbytes: 대규모 모델 효율적 로딩 및 양자화(Quantization) 지원
- torch (PyTorch): 딥러닝 프레임워크 (모델 학습, 추론 등)
- rank\_bm25 (BM25Okapi): 전통적인 정보 검색(문서 랭킹) 알고리즘 구현
- umap: 차원 축소 기법을 통한 데이터 시각화

## LangChain 생태계

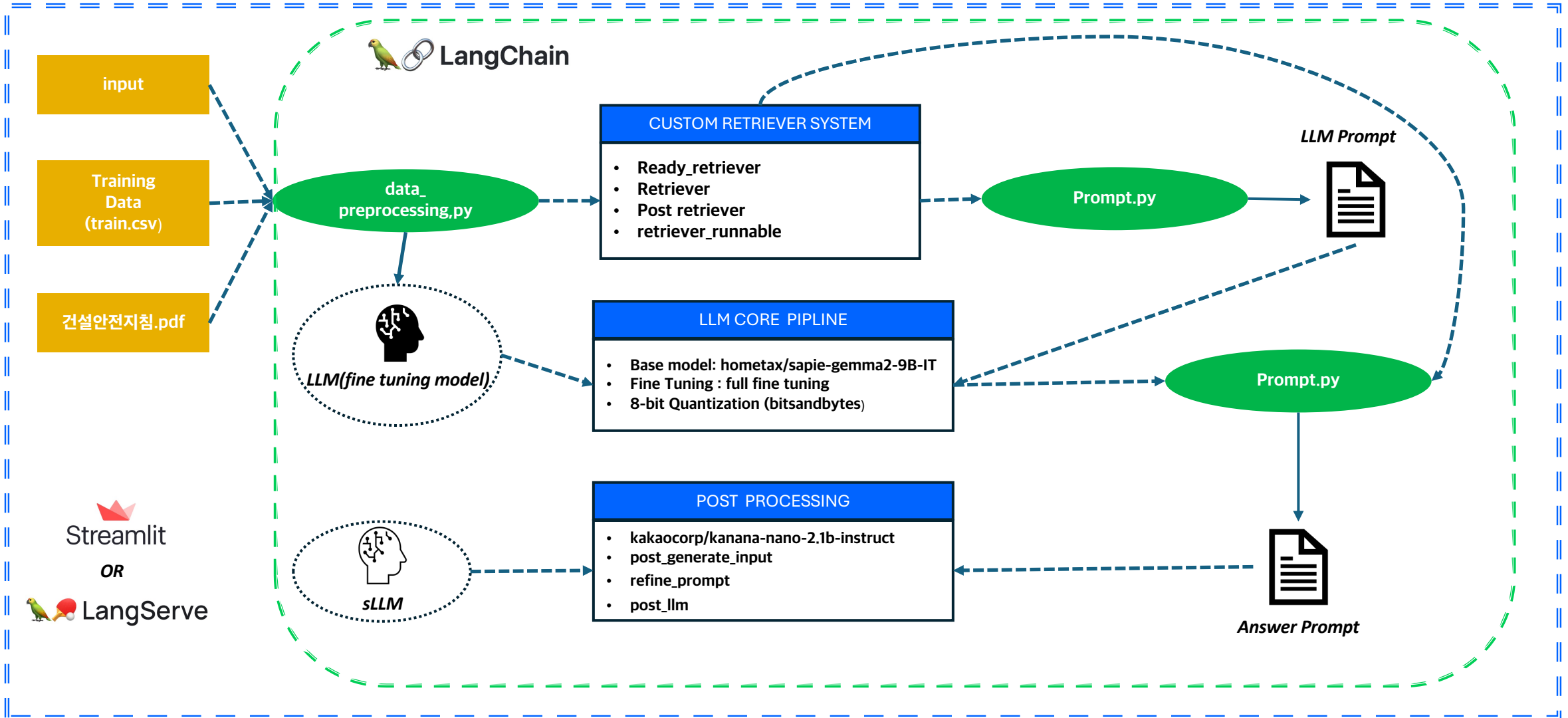
- langchain.embeddings, langchain.chains, langchain.prompts:
- LLM 기반 응용을 위한 체인 구성, 프롬프트 템플릿, 임베딩 생성
- langchain\_community.vectorstores (Chroma, FAISS):
- 벡터 스토어로 임베딩된 문서 검색 및 관리
- langchain\_core.runnables (RunnableMap, RunnableLambda, RunnableSequence, RunnablePassthrough):
- 파이프라인 구성 요소를 모듈화하여 실행할 수 있도록 지원

*Model Serving Process  
&  
Architecture*



# Model Serving Process & Architecture

## MODEL SERVING PROCESS FLOW



## 폴더 파일 구조

### (프로젝트 루트)

```
├── Utils/  
│   ├── data_preprocessing.py  
│   ├── Retriever.py  
│   └── prompts.py  
├── Evaluate/  
│   └── Matrix.py  
├── Training/  
│   ├── Train.py  
│   └── dataset_utils.py  
├── Data/  
├── Models/  
├── submissions/  
├── Model_Training.ipynb  
└── Project_Result.ipynb
```

## 주요 파이썬 파일 & 함수 설명

### 1. Utils/data\_preprocessing.py

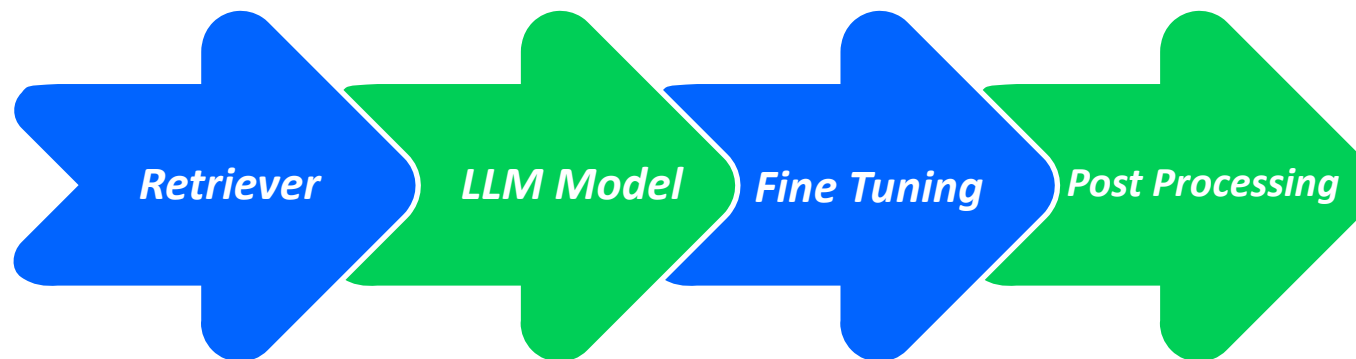
- data\_preprocessing(train, test)
  - Train / Test DataFrame을 받아서, 공사종류·공종·사고객체 등 텍스트 열을 대·중분류로 분리
  - 학습에 필요한 question, answer 필드를 생성해 DataFrame으로 반환
- insert\_word\_bridging\_rows(df, text\_col, bridging\_end, bridging\_start)
  - 텍스트를 일정 단어수 기준으로 앞뒤를 연결(브리징)하는 함수(페이지 연결 시 사용하는 예시)
- pdf\_to\_text(pdf\_directory, output\_file)
  - 지정한 폴더 내 PDF를 읽어 텍스트를 추출하고, CSV로 저장/반환

### 2. Utils/Retriever.py

- 전역 변수: bi\_encoder, cause\_embeddings, environment\_embeddings, answer\_embeddings, bm25\_cause, bm25\_env, train\_data
- ready\_retriever(model\_name, train\_df)
  - SentenceTransformer로 임베딩 모델 로드 → 사고원인·환경·답변 임베딩 계산 + BM25 세팅
- retriever(accident\_cause, accident\_environment, top\_k, alpha, beta, threshold)
  - 사고 원인·환경 텍스트 입력으로 BM25 + 코사인 유사도(임베딩)를 결합해 상위 유사 문서를 찾음
- post\_retriever(query\_text, top\_k, threshold)
  - 단순히 answer\_embeddings와 유사도가 높은 사례를 찾음
- retriever\_runnable(inputs)
  - LangChain Runnable 형태로 retriever를 감싸서 체인에서 활용 가능하도록 만든 함수

*Algorithm*

## Algorithm Process



01

### Retriever Selection

- 최적의 한국어 sentence embedding 테스트
- BM25 방식의 retriever 테스트
- Ensemble retriever 테스트

02

### LLM model Selection

- 개발 환경 최적화: 모델 크기, 추론 속도 등을 종합적으로 고려해 Base Model 후보 선정
- 대상 모델: Horangi 한국어 LLM 리더보드 상위 모델, 국내 기업에서 최근 배포한 LLM

03

### Fine Tuning

- hometax/sapie-gemma2-9B-IT모델을 기준으로 tuning을 진행
- Full fine tuning과 LoRA를 활용한 fine tuning 전략 중 모델과 데이터의 크기 및 학습 시간 등을 고려해 Full fine tuning 전략을 채택

04

### post processing model

- 후처리 모델의 경우 동작속도, GPU 메모리를 고려해서 상대적으로 작은 파라미터 크기의 sLLM 모델을 사용한다.
- 후처리 모델의 경우 정량적 평가보다는 정성적 평가를 진행해 모델을 선정했다.

## Retriever Selection

- 한국어 sentence embedding을 지원하는 모델 중 모델의 성능, 시간, 메모리 등을 고려해서 후보 모델을 선택
- 평가 방식은 train\_data에서 랜덤 추출로 validation data를 만들고 validation data의 사건 정보를 바탕으로 retriever를 진행한 뒤 추출된 문서의 target text와 validation data의 target text의 재발 방지 대책 및 향후 조치 계획에 대한 cosine 유사도도를 구한다.
- 추출은 top 3로 진행하며, 평가는 top1, top2, top3의 문서들이 대해서 유사도를 각각 구한다.
- [https://github.com/su-park/mteb\\_ko\\_leaderboard?tab=readme-ov-file](https://github.com/su-park/mteb_ko_leaderboard?tab=readme-ov-file) : 한국어 임베딩 모델 리더보드
- 더 나은 retriever의 판단 기준은 재발 방지 대책 및 향후 조치 계획의 코사인 유사도를 기준으로 판단하며 유사도 측정 모델은 'jhgan/ko-sbert-sts'로 진행한다.

## Embedding Model Result

- intfloat/multilingual-e5-large-instruct의 성능이 가장 좋게 도출 되었다.
- 크기가 작은, 중간, 큰 모델을 테스트 해봤지만 크기가 크다고 성능이 좋게 나오지는 않았다.

Model	Top1	Top2	Top3
jhgan/ko-sbert-nli	0.5624370054	0.5552473552	0.5621721343
Alibaba-NLP/gte-Qwen2-1.5B-instruct	0.5698127060	0.5620514323	0.5549694432
facebook/drama-1b	0.5614832385	0.5584166892	0.5570728830
intfloat/multilingual-e5-large-instruct	0.5717318398	0.5583943233	0.5585753730
upskyy/bge-m3-korean	0.5636812279	0.5560255649	0.5530657734
Salesforce/SFR-Embedding-2_R	0.5689209725	0.5621495386	0.5564089142

- 주어진 데이터의 특징으로 사고 환경에 대한 정보는 사건 마다 대부분 유사하다. 그래서 사고 환경 정보가 유사도를 측정하는데 있어서 많은 비중을 차지하는 것으로 보인다.
- 사고 원인의 경우 매우 다양하며 사고 원인에 따라 후속 조치가 많은 영향을 받는 것으로 판단되어 retriever를 할 때 두 정보의 가중치를 다르게하여 테스트를 진행했다.
- Accident enviroment(공사종류, 연면적, 층 정보, 인적사고, 물적사고, 공중, 사고객체)
  - Ex) 공사종류 대분류 '건축', 중분류 '건축물' 공중 대분류 '건축', 중분류 '철근콘크리트공사' 사고객체 '건설자재'(중분류: '철근')
- Accident\_cause(작업프로세스, 사고원인)
  - Ex) 작업 프로세스는 '설치작업'이며, 사고 원인은 '고소작업 중 추락 위험이 있음에도 불구하고, 안전난간대, 안전고리 착용 등 안전장치가 미흡하였음.'입니다

Embedding model (alpha: accident\_cause, beta: accident\_enviroment)

LM model	Top1	Top2	Top3
intfloat/multilingual-e5-large-instruct(Baseline)	0.568652	0.566165	0.561158
alpha_1.0_beta_0.0	0.568652	0.566165	0.561158
alpha_0.9_beta_0.1	0.573048	0.568066	0.559194
alpha_0.8_beta_0.2	0.575897	0.568070	0.558153
alpha_0.7_beta_0.3	0.576931	0.568909	0.555849
alpha_0.6_beta_0.4	0.573556	0.569862	0.559238

BM25 model (alpha: accident\_cause, beta: accident\_enviroment)

BM25	Top1	Top2	Top3
BM25 (Baseline)	0.566747	0.556489	0.549351
alpha_1.0_beta_0.0	0.572518	0.555372	0.552491
alpha_0.9_beta_0.1	0.572996	0.556817	0.552762
alpha_0.8_beta_0.2	0.577750	0.556057	0.554362
alpha_0.7_beta_0.3	0.578485	0.553591	0.558299
alpha_0.6_beta_0.4	0.575717	0.556977	0.554873

## Retriever ensemble Result (alpha:embedding model, beta: BM25 mdoel)

- 모델 구성 : 임베딩 모델 ( $\alpha$ ): 딥러닝 기반 semantic search, BM25 모델 ( $\beta$ ): 전통적 키워드 검색 알고리즘
- 가중치 변동 실험 :  $\alpha + \beta = 1.0$  조건으로 0.1 단위 조합 테스트, 모든 조합에서 동일 테스트셋 사용

Ensemble Model	Top-1	Top-2	Top-3
$\alpha=1.0, \beta=0.0$	0.576931	0.568909	0.555849
$\alpha=0.9, \beta=0.1$	0.578820	0.553504	0.561445
$\alpha=0.8, \beta=0.2$	0.578872	0.553436	0.560511
$\alpha=0.7, \beta=0.3$	0.578790	0.553129	0.559572
$\alpha=0.6, \beta=0.4$	0.578790	0.552954	0.559615
$\alpha=0.5, \beta=0.5$	0.578822	0.553216	0.558499
$\alpha=0.4, \beta=0.6$	0.578822	0.553341	0.557961
$\alpha=0.3, \beta=0.7$	0.578822	0.553344	0.558306
$\alpha=0.2, \beta=0.8$	0.578628	0.553634	0.558334
$\alpha=0.1, \beta=0.9$	0.578628	0.553367	0.558392
$\alpha=0.0, \beta=1.0$	0.578485	0.553591	0.558299

- 임베딩 모델에 80%, BM25에 20% 가중치 부여 시 가장 균형 잡힌 성능을 보입니다.
- 실제 서비스에서는  $\alpha=0.8/\beta=0.2$ 을 기본값으로 설정하고, 쿼리 특성에 따라 동적 가중치 조정이 가능하도록 아키텍처를 설계했습니다.

## Base Model Selection

- 개발 환경 최적화: 모델 크기, 추론 속도 등을 종합적으로 고려해 Base Model 후보 선정
- 대상 모델: Horangi 한국어 LLM 리더보드 상위 모델, 국내 기업에서 최근 배포한 LLM
- 평가 방법 : 재발방지대책 및 향후조치계획(Target) 적합성, 생성 문장의 품질 (정량/정성 평가 병행)
- 평가 프로세스: 1차 정성 평가: 샘플 문장에 대한 인간 평가 (유창성, 일관성, 목적 적합성 등), 2차 정량 평가: 객관적 지표를 통한 성능 측정

모델 이름	Cosine 유사도 평균	Final Score	200개 샘플 추론 시간	비고
recoilme/recoilme-gemma-2-9B-v0.4	0.56591845	0.3998	53분	9B 모델
Qwen/QwQ-32B-Preview	0.5654715	0.4082	62분	
hometax/sapie-gemma2-9B-IT	0.5703157	0.4226	41분	9B 모델
LGAI-EXAONE/EXAONE-3.5-32B-Instruct	0.58709943	0.4306	30분	32B 모델, 최고 성능

## 최종 모델 선정 배경

- LGAI-EXAONE/EXAONE-3.5-32B-Instruct 모델: Cosine 유사도(0.587)와 Final Score(0.4306)에서 최고 성능 기록 그러나 경진대회 제약사항 위반으로 사용 불가
- hometax/sapie-gemma2-9B-IT 모델 선정 이유: 대회 규정 준수 가능 모델 중 최상위 성능, 32B 모델 대비 성능 차이 미미( $\Delta$  Cosine 유사도 0.017), 모델 크기 대비 효율적인 성능 발휘

## 평가 과정에서의 주요 발견

- 모델 크기와 성능의 상관관계: 32B 모델이 9B 모델 대비 약 2.9% 우수한 성능 그러나 성능 차이가 크지 않아, 규모 확대의 효용성에 의문
- 포괄적인 테스트 결과: 표에 제시되지 않은 다수 모델 테스트 수행 -> 정성 평가(유창성, 일관성 등)에서 대부분 미흡한 결과 도출
- 이러한 분석을 바탕으로, hometax/sapie-gemma2-9B-IT를 베이스 모델로 선정하였으며, 튜닝을 통해 성능을 극대화

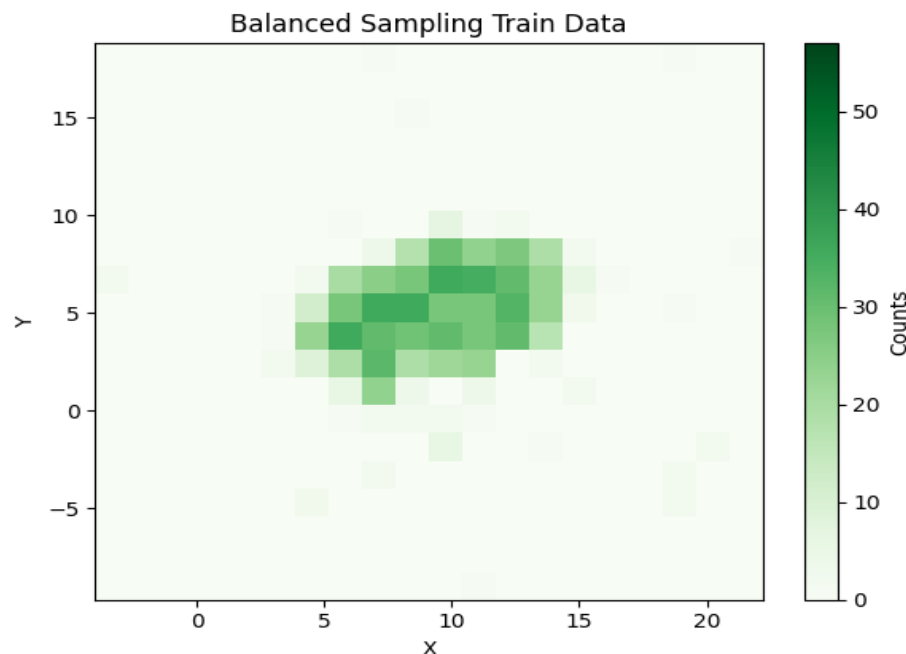
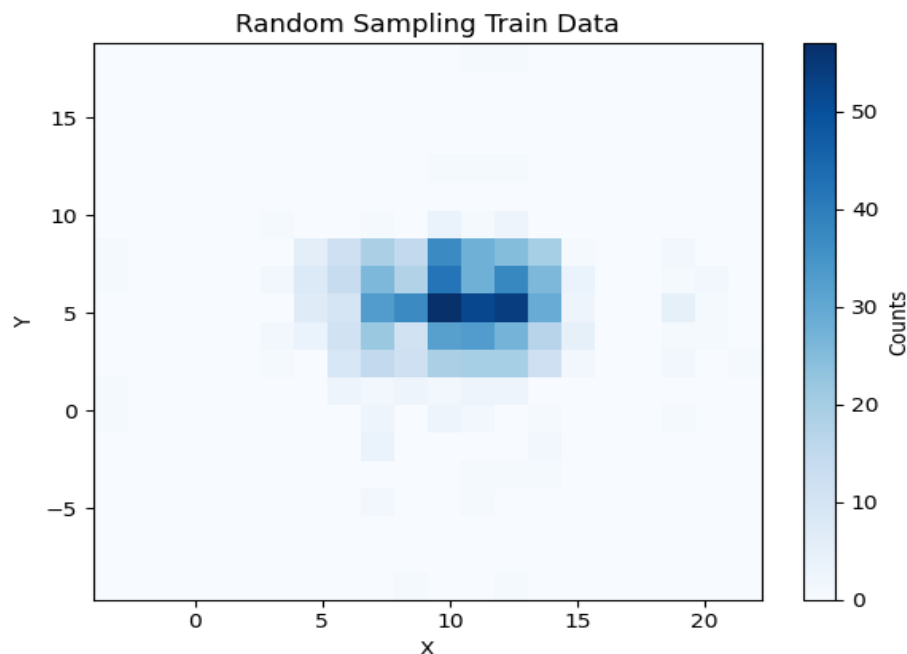


## Model tuning 전략

- hometax/sapie-gemma2-9B-IT모델을 기준으로 tuning을 진행
- Full fine tuning과 LoRA를 활용한 fine tuning 전략 중 모델과 데이터의 크기 및 학습 시간 등을 고려해 Full fine tuning 전략을 채택

## 데이터 활용

- 건설안전지침(PDF) 데이터와 train data를 활용하여 모델 튜닝을 진행하였다. 건설안전지침(PDF)은 표지와 목차를 제외한 본문 전체를 사용하였으며, train data는 epoch마다 1,000개의 데이터를 샘플링하여 학습을 수행하였다.
- 건설안전지침(PDF) 데이터의 경우 pdf를 텍스트화 하는 PyMuPDF, PyPDF2, pdfminer.six, pdfplumber 라이브러리를 테스트 한 결과 pdfplumber를 사용했을때 text 추출이 가장 좋았습니다.
- Train data에 대해 무작위(random) 샘플링 방식을 사용할 경우 특정 target의 분포가 일부 의미에 편중되어 학습이 왜곡되는 현상이 나타났다. 이를 해결하기 위해, target을 임베딩한 후 차원 축소를 수행하여 2차원 좌표 평면에 표현하였다. 이후 좌표 평면을 여러 구간으로 나누고 각 구간에서 고르게 데이터를 샘플링하여 특정 target에 편향되지 않도록 균형 잡힌 학습 데이터를 구성하였다.



## Full Fine Tuning Result

- 건설안전지침(PDF) 데이터와 train data를 활용하여 모델 튜닝을 진행하였다
- Epoch 3에서 최고의 성능(Final Score: 0.40115)을 기록하였으며, 이 시점을 기준으로 모델이 최적의 상태에 도달한 것으로 판단됩니다.
- Epoch 4부터 성능 저하가 관찰되기 시작, 모든 평가 지표에서 점진적인 악화 추세(Jaccard Score가 0.05666(Epoch3) → 0.04235(Epoch5)로 24.8% 감소)
- Epoch 별 파인튜닝 모델의 성능 평가는 Retriever 없이 단일 사건 정보를 입력하여 얻은 결과를 기반으로 진행한다.

Epoch	Final Score	Cosine Score	Jaccard Score	학습 현상
0	0.33125	0.45535	0.04169	초기 수렴
1	0.39262	0.53399	0.06275	급격한 향상
2	0.39404	0.53571	0.06347	미세 조정
3	0.40115	0.54880	0.05666	최적화 진입
4	0.39982	0.54712	0.04821	과적합 시작
5	0.39876	0.54533	0.04235	값 편향 심화

## Retriever 유무 비교 평가

- Retriever를 활용한 유사 사건 정보 포함 여부에 따른 성능 비교
- Retrieval 기반 모델이 Non-Retriever 대비 모든 지표에서 우수 (Final Score +5.5%)

Epoch3 Model	Final Score	Cosine Score	Jaccard Score
With retriever	0.43759	0.5853452	0.09284
Non retriever	0.41482	0.5626778	0.06981

## Post processing

- LLM에서 나온 초안을 사용자 친화적인 답변으로 다듬기 위해, sLLM을 활용한 후처리 과정을 적용
- 동작 속도와 GPU 메모리 사용량을 고려해 상대적으로 파라미터가 작은 sLLM 모델을 사용하였으며, 후처리 모델은 정량적 지표보다는 정성적 평가를 거쳐 최종 선정.
- 최종적으로 kakaocorp/kanana-nano-2.1b-instruct 모델을 채택
- 후처리 시에는 LLM의 결과와 유사한 형태의 답안(Train 데이터에서 Retrieve)도 함께 입력하여, 기존 답변 형식을 참고하도록 설계했습니다.

## Example

index	LLM result sentence	Post processing sentence
5	안전발판 설치와 작업절차 준수에 대한 재발 방지 대책과 향후 조치 계획. 재발방지대책 및 향후조치계획: 현장 정리정돈과 근로자 안전교육의 철저함을 통	작업발판 설치와 작업자 안전교육 강화를 통한 재발 방지 대책 및 향후 조치 계획.
6	매설작업 시 신호수 배치를 통한 작업 추진과 안전 교육 실시.	매설작업 시 신호수 배치를 통한 작업 추진과 안전 교육 실시.
7	작업 전 TBM을 통한 작업순서 및 안전작업 방법 교육 실시, 작업자 안전교육 및 TBM을 통한 안전수칙 준수, 작업자 안전교육 및 TBM을 통한 개인보호구 착용 철저, 안	작업 전 TBM을 통한 작업순서 및 안전작업 방법 교육 실시, 작업자 안전교육 및 TBM을 통한 안전수칙 준수, 작업자 개인보호구 착용 철저, 작업 시작 전 작업순서와 방법 철저 숙지, 안전수칙 준수
8	작업자 안전교육 실시와 안전관리 철저에 대한 재발 방지 대책과 향후 조치 계획.	작업자 안전교육 및 안전관리 철저에 대한 재발 방지 대책과 향후 조치 계획.
9	작업 전 TBM을 통한 작업순서 및 안전작업절차 교육 실시, 해체작업 시 안전대 걸이시설 설치 및 착용 확인, 작업자 안전교육 실시, 해체작업 시 충돌방지 조치 및 작업	작업 전 TBM을 통한 작업순서 및 안전작업절차 교육 실시, 해체작업 시 안전대 걸이시설 설치 및 착용 확인, 작업자 안전교육 실시, 해체작업 시 충돌방지 조치 및 작업자 보호 조치 강화.
10	고령 작업자 및 건강상태 불량자에 대한 현장 작업 금지 지시.	고령 작업자 및 건강상태 불량자에 대한 현장 작업 금지 지시.

*Thank You*