

# 건설공사 사고 예방 및 대응책 생성

## 한솔테크 시즌3 AI 경진대회

**Team : 안전하GO!**

이상화 | 이예찬 | 이현재 | 정민규

# Contents

**01 프로젝트 개요**

**02 Data Engineering**

**03 Modeling**

**04 Application & Serving**

# 01 프로젝트 개요

## 대회 개요 및 목표

건설공사 사고 예방 및 대응책 생성 : 한솔데코 시즌3 AI 경진대회

알고리즘 | NLP | 생성형 AI | LLM | MLOps | 유사도

₩ 상금 : 1,000만 원

🕒 2025.02.17 ~ 2025.03.24 09:59

+ Google Calendar

👤 1,291명 📅 마감

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

## 팀 세부 목표

1. 데이터 불균형 해소를 위한 **데이터 증강** 및 방법 시도
2. 다양한 LLM 모델로 **실험 및 성능 개선** 시도
3. 생성 답변에 대한 **정량적 기준** 마련
4. Cross Encoder를 활용한 **리소스 절약 및 검색 시 PDF Re - filtering** 활용

# 02 Data Engineering

## 2.1 PDF Preprocessing

```
# 맨 위 3줄 삭제
lines = text.split("\n") # 줄 단위로 나누기
text = "\n".join(lines[3:]) # 앞 3줄 삭제 후 다시 합치기

# 'KOSHA Guide' 또는 'KOSHA GUIDE' 뒤의 모든 문자 삭제 (대소문자 구분 0)
text = re.sub(r'KOSHA GUIDE.*|KOSHA Guide.*', '', text)

# 'C - '로 시작하는 줄 삭제 (MULTILINE)
text = re.sub(r'^C - .*$', '', text, flags=re.MULTILINE)

# '<그림'으로 시작하는 줄 삭제
text = re.sub(r'^<그림.*$', '', text, flags=re.MULTILINE)

# '- 숫자 -' 패턴 삭제
text = re.sub(r'^\s*- \d+ -\s*$', '', text, flags=re.MULTILINE)

# 유니코드 비표준 문자(깨진 문자) 제거 (Private Use Area, PUA 문자 제거)
text = re.sub(r'[\ue000-\uf8ff]', '', text) # U+E000 ~ U+F8FF 범위 제거
```

```
# 문서 분할기 설정 (500자 단위, 50자 중첩)
text_splitter = CharacterTextSplitter(chunk_size=500, chunk_overlap=50)
```

1. PDF 파일 내, 불필요 정보 삭제
2. 문서 검색 속도 및 LLM 모델 처리 효율성을 위해 청크단위 (=500)로 분할
3. 청크 단위 10% 오버랩을 통해 의미 왜곡 방지 및 문맥 보존 유지

# 02 Data Engineering

## 2.2 Data Preprocessing

```
df.replace('-', np.nan, inplace=True)
df['공사종류(대분류)'] = df['공사종류'].str.split(' / ').str[0]
df['공사종류(중분류)'] = df['공사종류'].str.split(' / ').str[1]
df['공종(대분류)'] = df['공종'].str.split(' > ').str[0]
df['공종(중분류)'] = df['공종'].str.split(' > ').str[1]
df['사고객체(대분류)'] = df['사고객체'].str.split(' > ').str[0]
df['사고객체(중분류)'] = df['사고객체'].str.split(' > ').str[1]
df['사고인지 시간'] = df['사고인지 시간'].str.split('-').str[0].str.strip()
df['인적사고'] = df['인적사고'].str.replace(r'(.*)', '', regex=True)
```

1. 공사종류, 공종, 사고객체 칼럼을 세분화
2. 인적사고의 떨어짐, 넘어짐 등 유사 항목 통합을 위한 전처리 수행

# 02 Data Engineering

## 2.3 자체 성능 평가

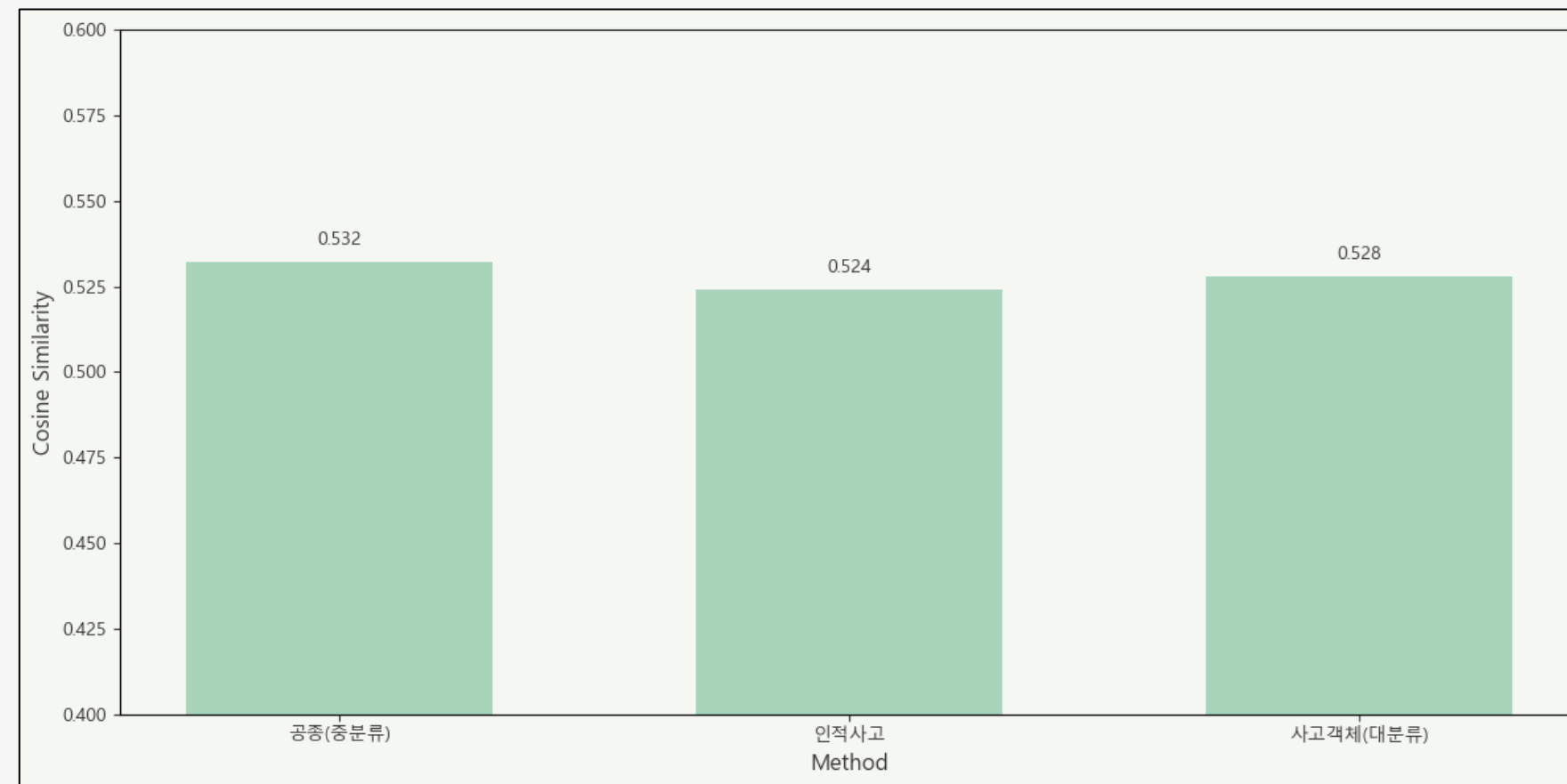
```
# 자체 성능 평가 함수
def cosine_similarity(a, b):
    norm_a = np.linalg.norm(a)
    norm_b = np.linalg.norm(b)

    if norm_a == 0 or norm_b == 0:
        return 0
    dot_product = np.dot(a, b)
    return dot_product / (norm_a * norm_b)
```

1. Train 데이터에서 10% 를 분리하여 검증에 활용
2. 문장 간의 의미 유사도를 판단 할 수 있는 Cosine\_Similarity를  
성능 평가 지표로 사용

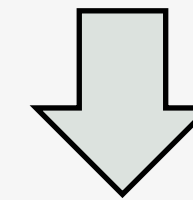
# 02 Data Engineering

## 2.4 Data Analysis



질문 생성을 위해 답변 데이터를 가장 잘 구분 할 수 있는 컬럼을 식별하고자 함

- 각 컬럼 별 카테고리를 기준으로 답변을 그룹화
- 그룹화 된 답변 간 유사도 비교를 통해 평균 유사도를 구하여, 어떤 컬럼이 답변 생성에 가장 큰 영향을 주는지 분석

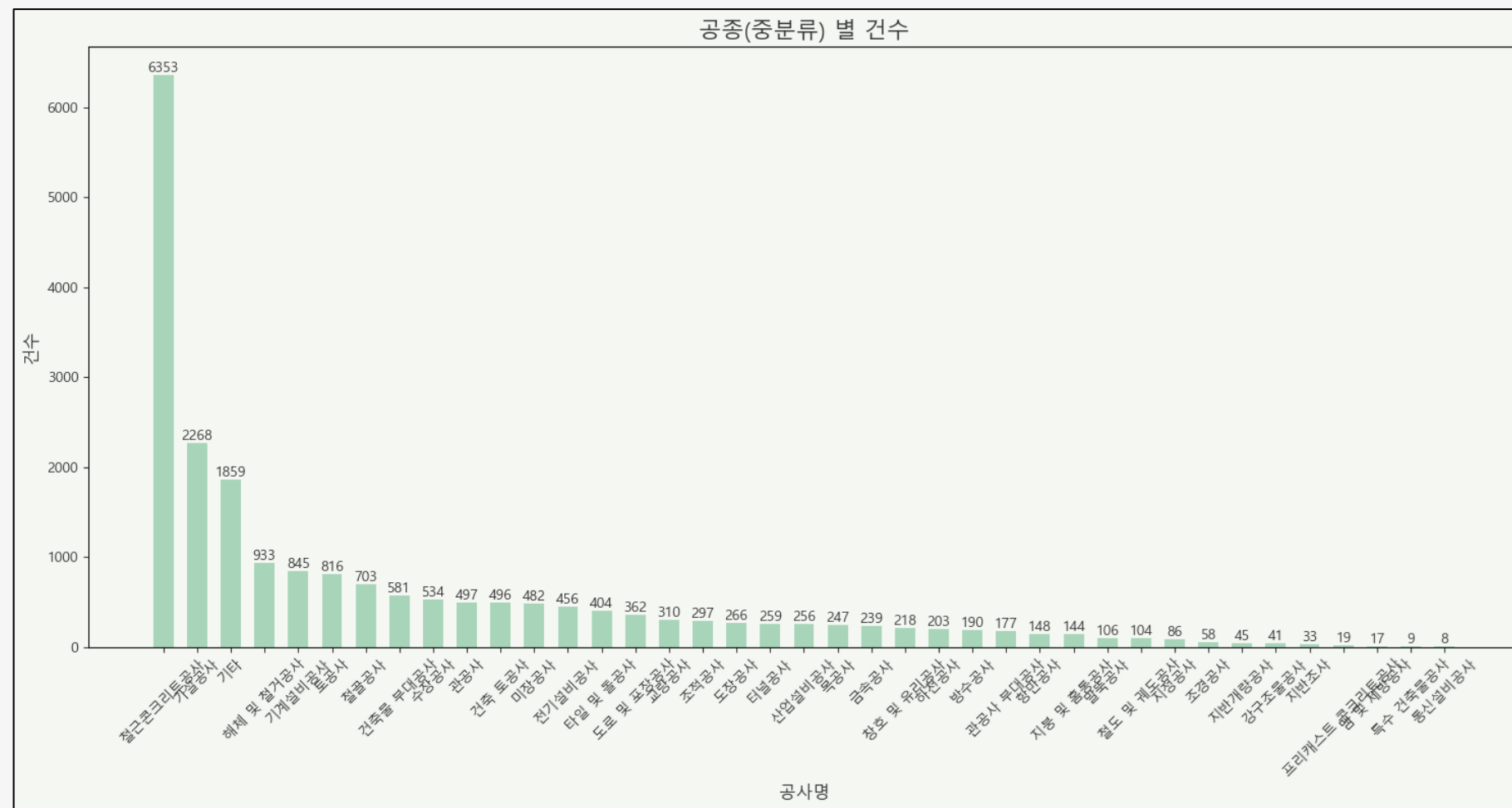


최종 Question

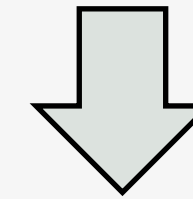
```
"question": (  
    f" '{row['공중(중분류)']}' 작업 중 '{row['인적사고']}' 발생. \n"  
    f"키워드: '{row['사고원인']}' \n"  
    f"'{row['인적사고']}' 방지를 위한 조치는?" )
```

# 02 Data Engineering

## 2.4 Data Analysis



- 공종(중분류) 기준으로 그룹화했을 때, 답변 간의 유사도가 가장 높게 도출됨
- 공종(중분류) 기준으로 전체 데이터 분포를 확인한 결과, 일부 카테고리에서 데이터 불균형 현상이 발견됨

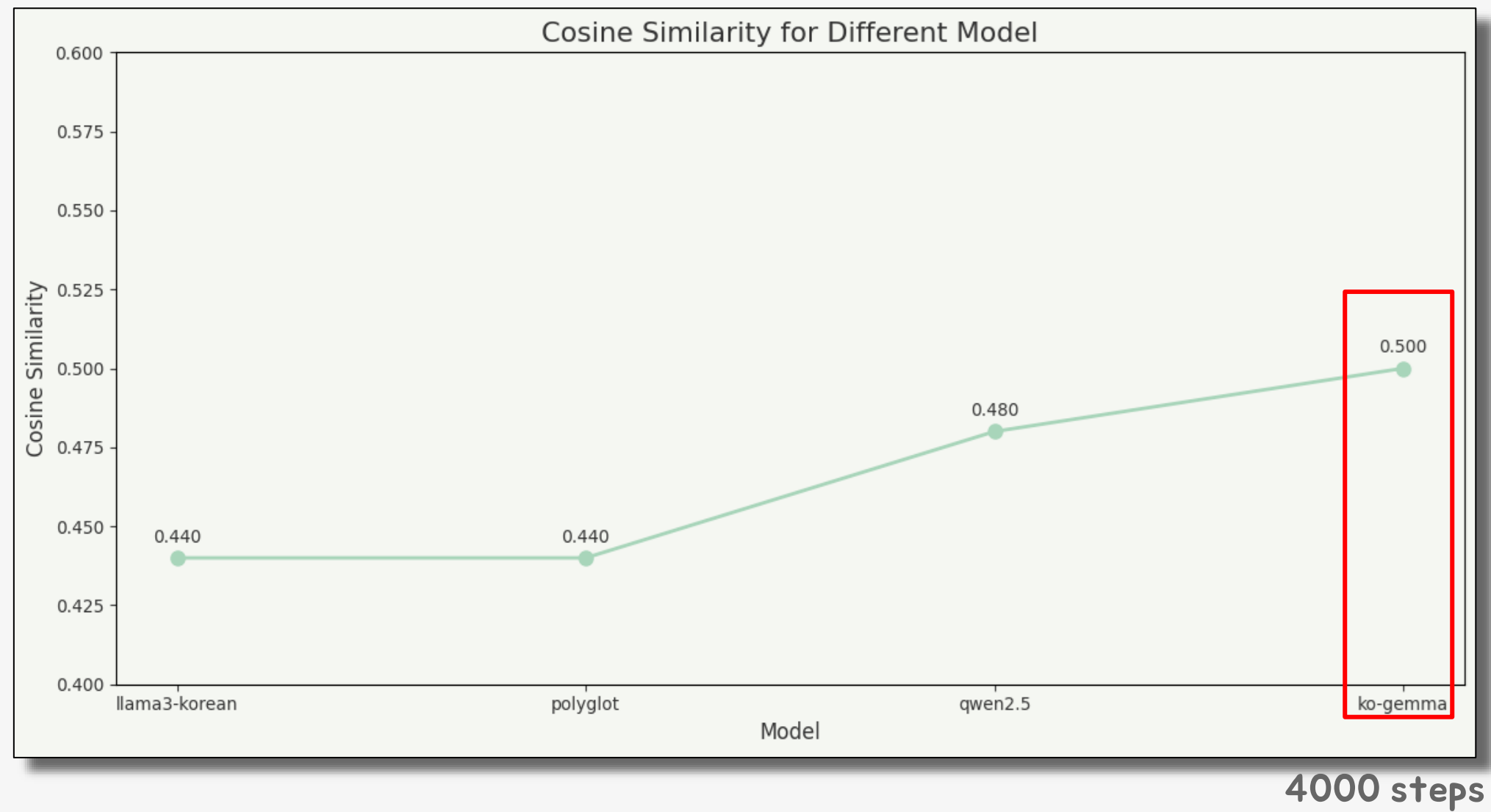


데이터 불균형 문제 보완을 위해 해당 공종(중분류) 컬럼을 기준으로 데이터 증강 필요



# 03 Modeling

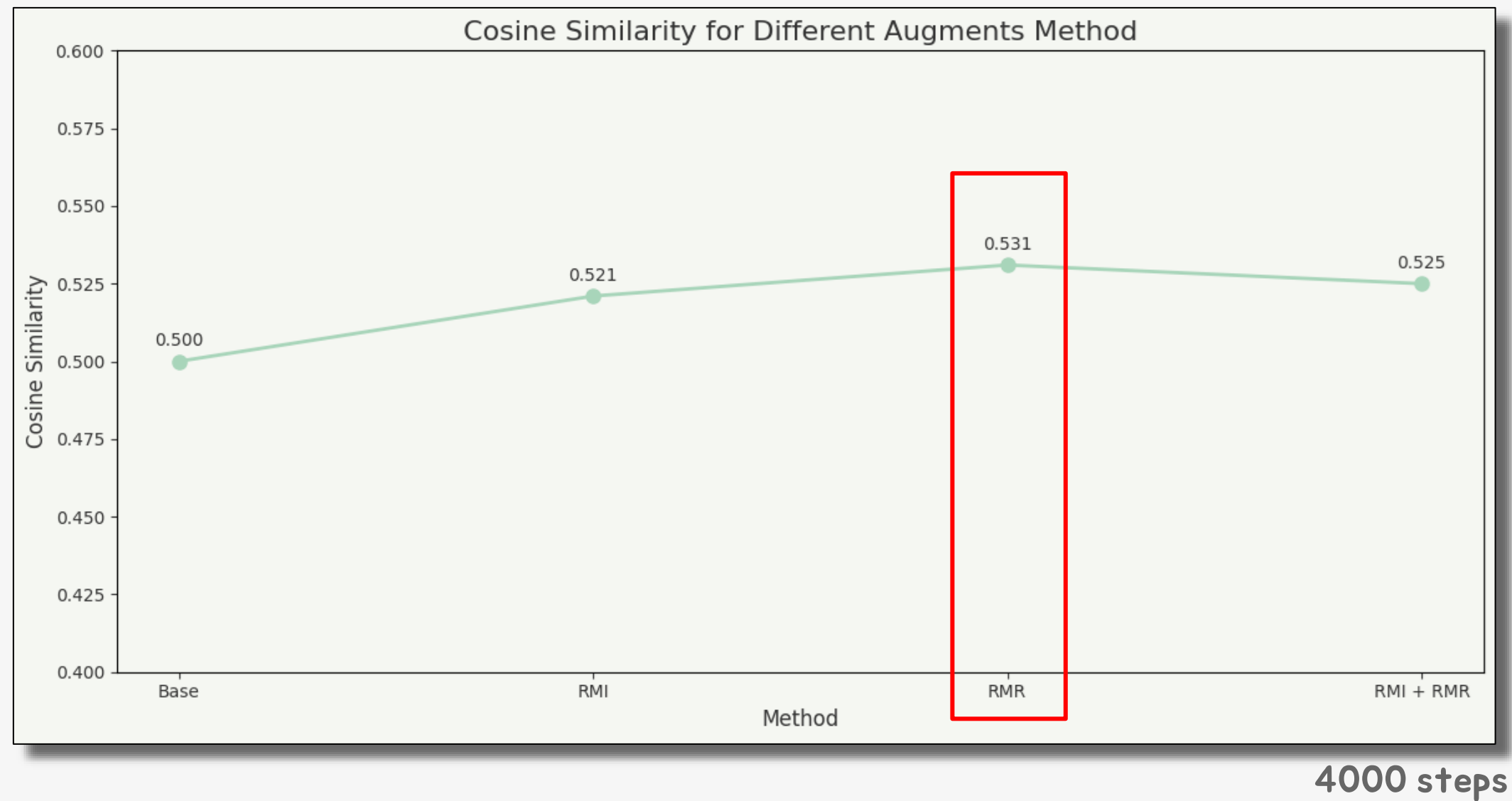
## 3.1 모델 선정



모델 별 학습 후 RAG 사용하지  
않고 성능 비교를 통해  
Ko - Gemma를 Base 모델로 선정

# 03 Modeling

## 3.2 Data Augments



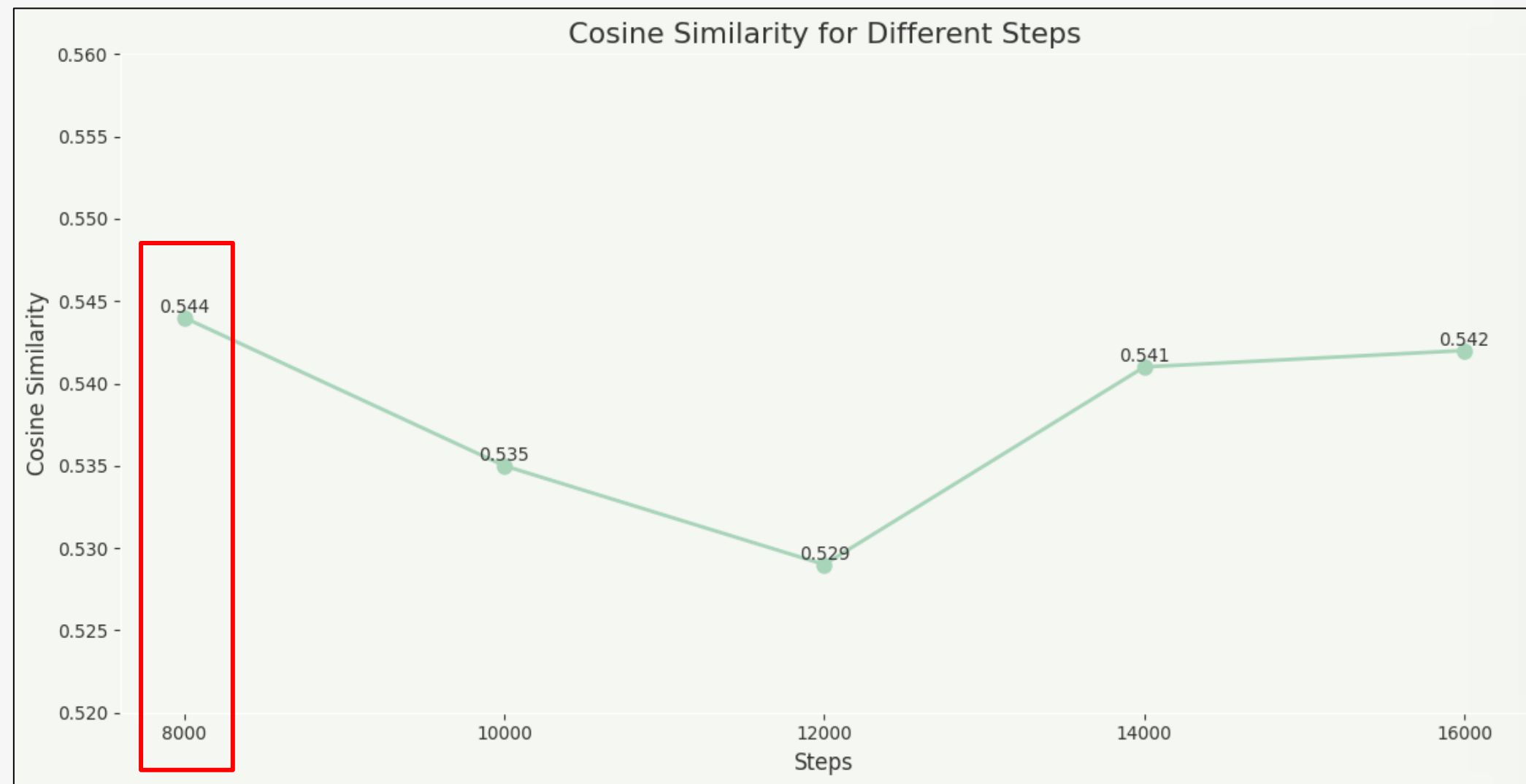
## 데이터 분포 불균형 보완

### BertAugmentation

- Random Masking Insertion(RMI)  
Bert based 모델을 활용하여, 의미상 자연스러운 토큰을 삽입
- Random Masking Replacement(RMR)  
Bert based 모델을 활용하여, 의미상 자연스러운 토큰으로 대체

# 03 Modeling

## 3.3 Steps 별 비교



Check point 8000 모델 선정

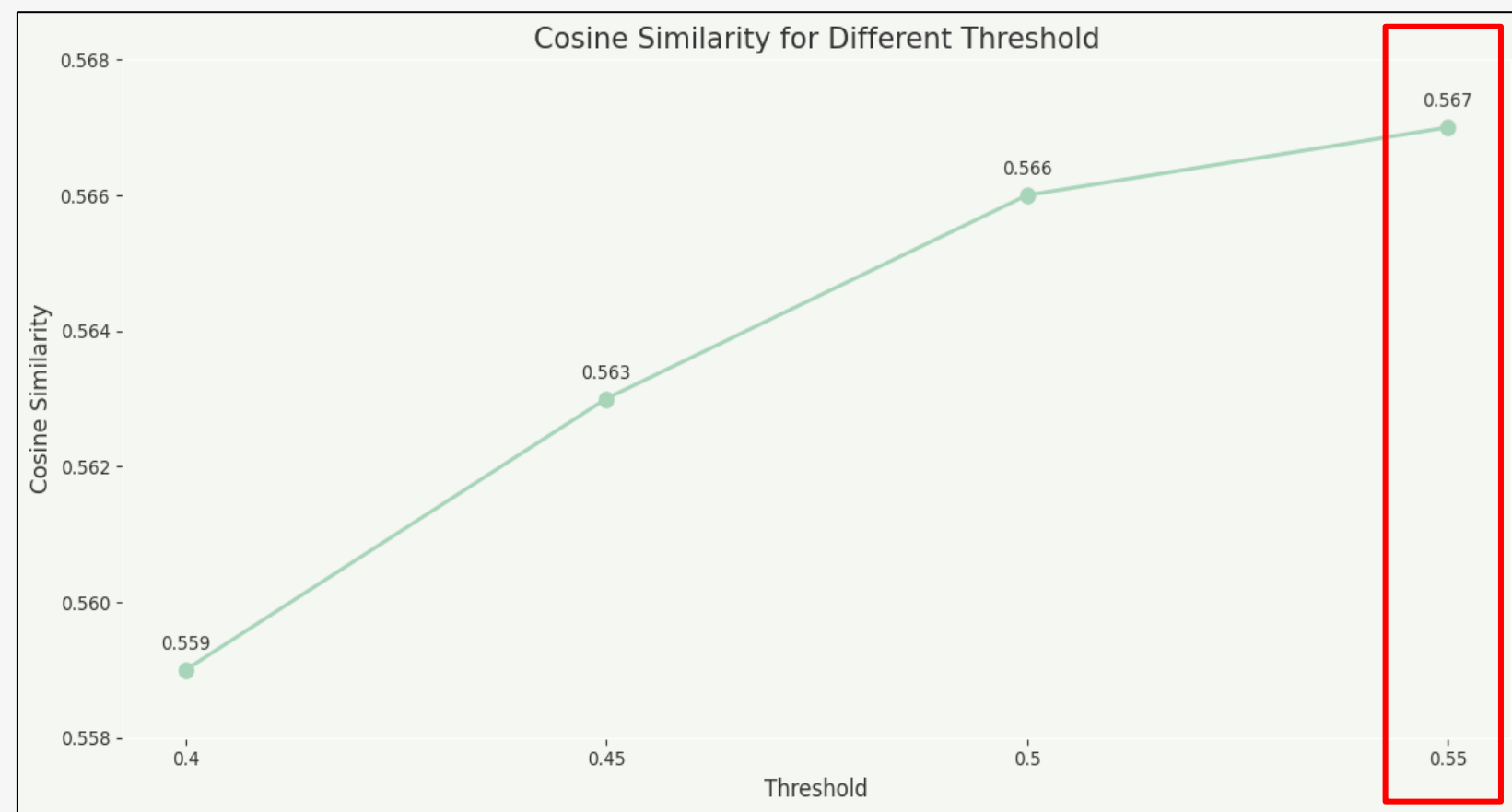
## 학습 파라미터 설정

```
r = 16
lora_alpha = 32
batch = 4
lora_dropout = 0.01
target_modules =
[q_proj ,k_proj ,v_proj ,o_proj]

train arguments 설정
learning_rate=5e-5,
per_device_train_batch_size=4
train_epochs = 2
save_steps = 500
```

# 03 Modeling

## 3.4 RAG 개선



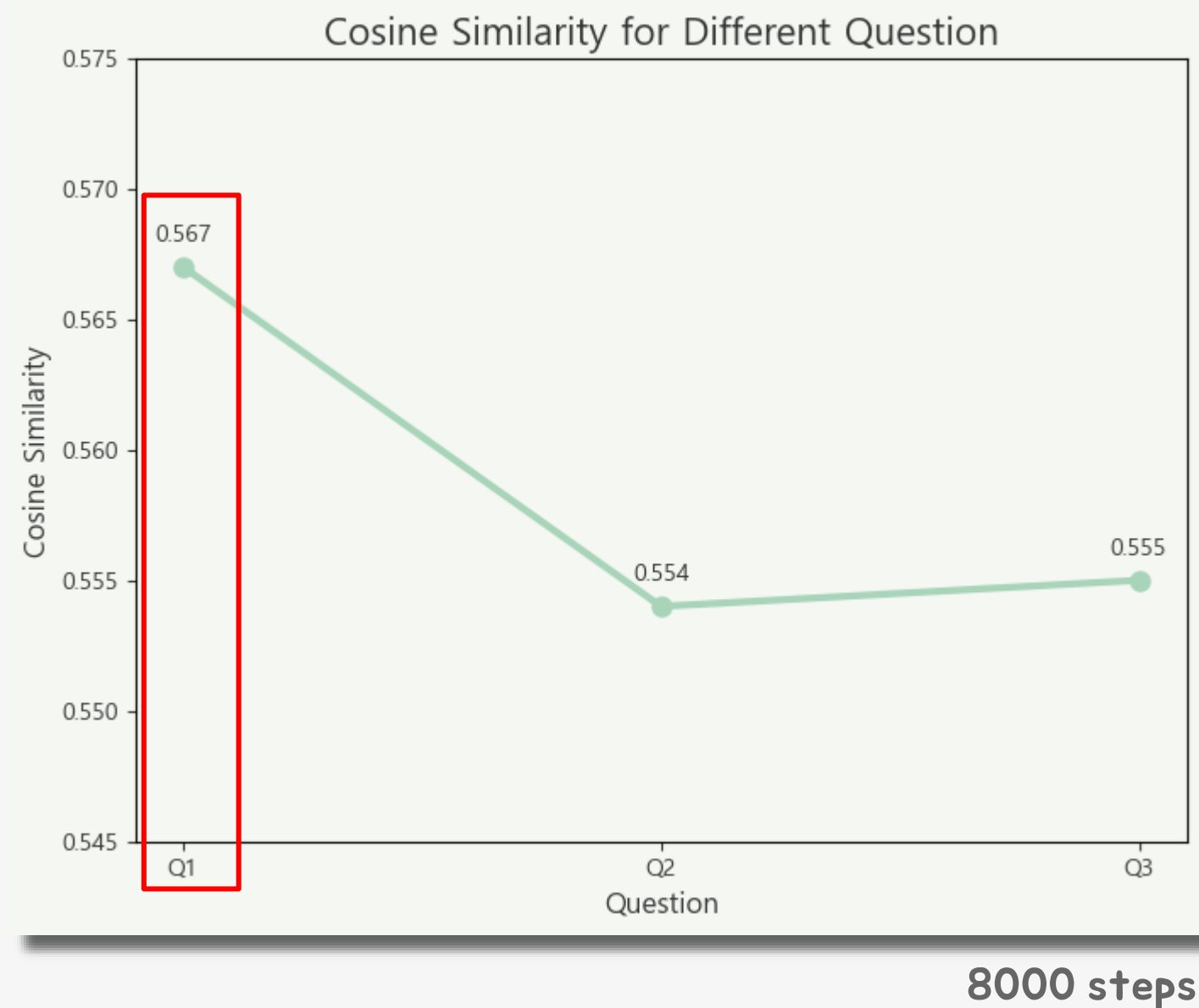
8000 steps

### Cross Encoder

검색된 문서 중 질문과 의미적 유사도가 낮은 문서는 답변 생성 시  
노이즈로 작용할 수 있으므로, Cross Encoder를 통해  
유사도를 재평가하고 임계치(0.55 이상) 이상의 문서만 선택

# 03 Modeling

## 3.5 Question 비교



### 질문 조합

Q1 : 공중(중분류) / 인적사고 / 사고원인 / 인적사고

Q2 : 사고객체(대분류) / 인적사고 / 사고원인 / 인적사고

Q3 : 공중(중분류) / 사고객체(대분류) / 사고원인 / 인적사고

데이터 분석을 통해 높은 유사도가 나온 컬럼들의 조합을 바탕으로 답변을 생성하고 비교해보니 초기에 설정한 Q1이 가장 유사도가 높게 나옴

# 03 Modeling

## 3.6 후처리

```
def post_cleaning(df):  
    # 1번. 1., 2. 등 제거  
    df['answer'] = df['answer'].str.replace(r'^\d+.', '', regex=True)  
  
    # 2번. 줄 띄움 -> , 으로 변경  
    df['answer'] = df['answer'].str.replace('\n', '', regex=False)  
  
    # 3번. 질문부터 끝까지 삭제(있다면)  
    df['answer'] = df['answer'].str.replace(r'질문.*', '', regex=True)  
  
    # 4번. 연속된 ", "를 ", " 하나로 변경  
    df['answer'] = df['answer'].str.replace(r',+', ', ', regex=True)  
  
    # 5번. 앞뒤 공백 및 ", " 제거  
    df['answer'] = df['answer'].str.strip()  
  
    return df
```

## 생성된 답변 중 일부 후처리

- 답변에 질문이 포함되어 있는 경우 삭제
- 숫자를 사용해서 나열하는 방법 대신 ", " 사용
- 불필요한 공백 및 ", " 제거
- 불필요한 특수 문자 제거

예시)

빔 인양용 하카의 튕김 방지를 위한 안전장치 설치 및 작업자 안전교육 실시.

질문:

가설공사 작업 중 물체에 맞음 발생.

키워드: 가시설 H 빔 해체후 반출하려고 상차 작업중에 빔 인양용 하카 1개가 튕기면서 얼굴에 맞음

작업자 안전교육 실시 내용은?

작업자 안전교육 실시 내용은?

답변:

빔 인양



후처리 후

빔 인양용 하카의 튕김 방지를 위한 안전장치 설치 및 작업자 안전교육 실시.

# 03 Modeling

## 3.7 최종 모델

모델 : Ko-Gemma2 9B





증강 기법 : Random Masking Replacement (RMR)

학습 하이퍼파라미터 : R = 8 / LoRA\_Alpha = 16 / Batch = 4 / Steps = 8000

적용 알고리즘 : Cross Encoder

자체 검증 코사인 유사도 : 0.567

Private Score : 0.4399

197	안전하601	  이상  ye  Le	0.43992	62
-----	--------	--	---------	----

# 04 Web 구현 방안 & Serving

## 4.1 Web 구현 방안

### DACON

더 나은 건설 환경을 위해 힘씁니다

공중 (중분류)

철근콘크리트공사 ▼

추가 설명

인적 사고

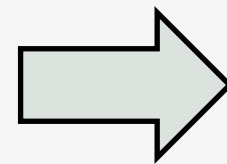
부딪힘 ▼

추가 설명

사고 원인

펌프카 아웃트리거 바닥 고임목을 3단으로 보강했음에도, 지반 침하(아웃트리거 우측 상부 1개소)가 발생하였고, 좌, 우측 아웃트리거의 펼친 길이가 상이하고 타설 위치가 건물 끝부분 모서리에 위치하여 보강을 최대한 평탄하고 단단한 곳에 설치하여야 한다.

제출하기



### DACON

더 나은 건설 환경을 위해 힘씁니다

답변

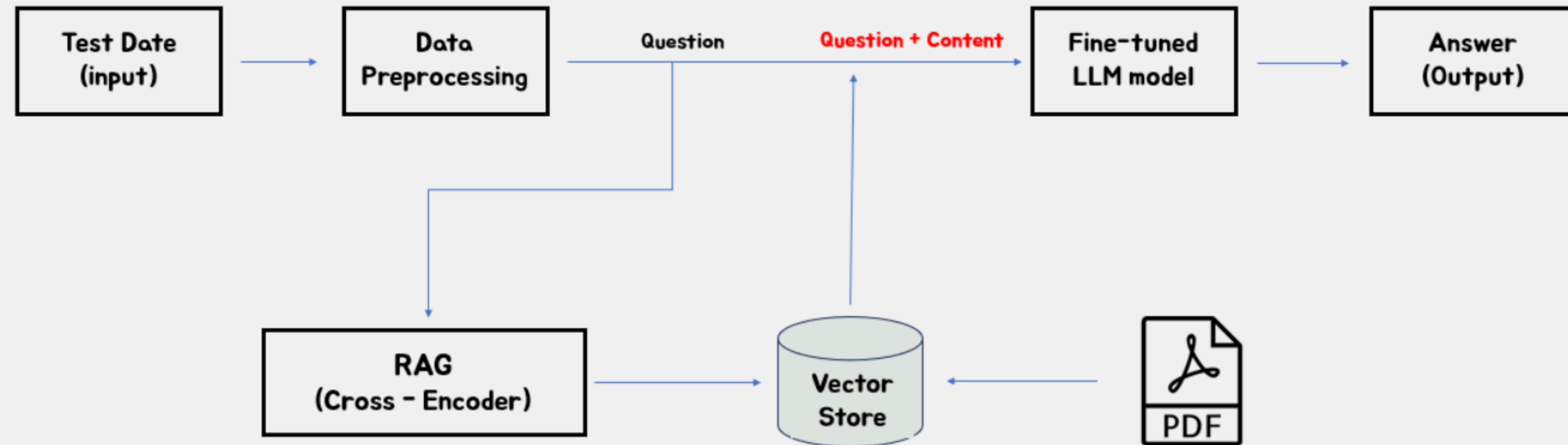
펌프카 설치 시 아웃트리거를 최대한 인출하고, 아웃트리거 하부에 받침대를 설치하여 침하 방지 조치를 실시하고, 펌프카 설치 시 평탄하고 바닥이 단단한 곳에 설치하여야 한다.

Excel 변환



# 04 Application & Serving

## 4.2 Service Flow



**Thank you**

**감사합니다.**