

## 목차

1. 프로젝트 개요
  - a. 배경 및 필요성
  - b. 최신 동향 및 한계점
2. 프로젝트 수행 절차 및 방법
  - a. 시스템 개요
  - b. 챌린지
3. 시스템 디자인
  - a. Solar
  - b. 평가 시스템
  - c. 시스템 연결
  - d. 목소리 및 영상 합성
4. 프로젝트 수행 결과
  - a. Solar 실험 결과
  - b. 평가 시스템 실험 결과
  - c. 전체 프로젝트 결과
5. 자체 평가 의견
  - a. 결론
  - b. 추후 고도화 방향
6. 프로젝트 팀 구성 및 역할
  - a. 프로젝트 타임라인
  - b. 역할

# 1. 프로젝트 개요

## a. 배경 및 필요성

### 1) 캔슬 컬쳐

오늘날 1인 미디어가 급증하며, 유튜버들도 기업 수준의 수익을 창출하고 있다. 하지만, 캔슬 컬쳐라는 단어가 있듯, 한순간의 발언 실수로 인해 채널이 위태로워지는 상황이 발생한다. 또한 직접적으로 논란이 되지 않더라도, 유튜브는 자체 알고리즘을 통해 논란 가능성이 있는 영상의 노출을 제한하고 있다.

뉴스에서도 앵커가 논란성 발언을 하여 해당 부분만 재녹화함으로서 뉴스의 신뢰도 논란이 발생한 사례가 있다.

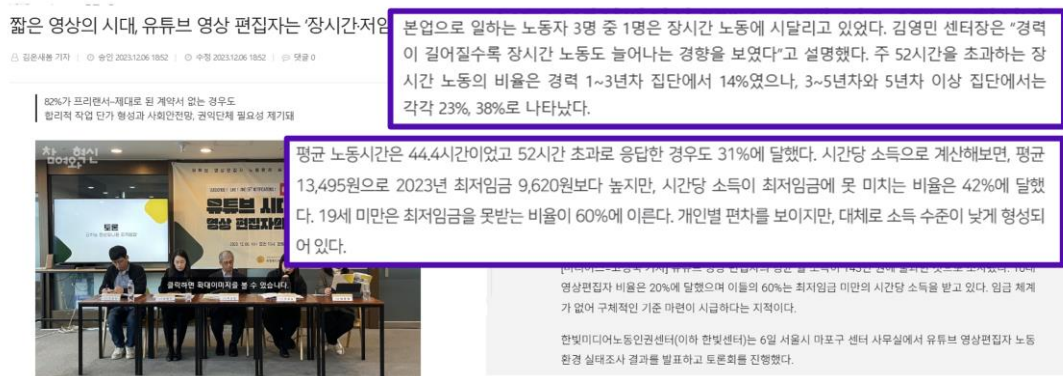
즉, 영상이 배포되기 전, 발언 위험도를 평가하고 수정하는 일은 매우 중요하다.



### 2) 캔슬컬쳐와 AI 도입의 필요성

영상편집 중 민감발언을 사람이 수작업으로 찾아 보면 편집 시간이 길어진다는 문제가 존재한다. 더욱 심각한 점은, 판단 기준이 제각각이기 때문에 편집자의 주관에 따라 놓치는 부분이 존재할 수 있다는 것이다.

따라서 AI 기반의 발언 위험도 평가 및 동영상 편집 시스템을 제안하고자 한다.



## b. 최신 동향 및 한계점

### 1) 민감 발언 탐지 모델

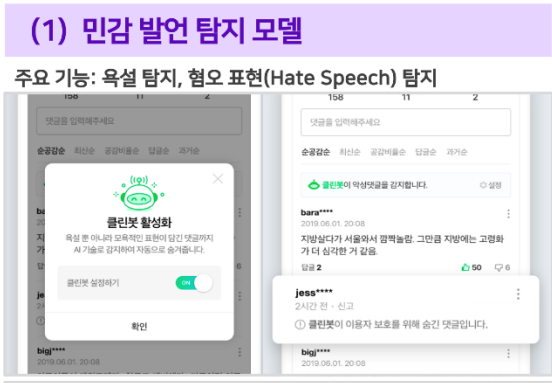
민감 발언 탐지 모델은 네이버 클린봇과 같이 일상에서도 흔히 찾아볼 수 있으나 대부분 직접적인 욕설 탐지에만 맞추어져 있으며, 미묘한 민감 발언 탐지 성능은 미흡하다.

### 2) AI 기반 영상 편집 프로그램

영상 편집에 AI를 접목하여 편집 시간 단축을 도모하고 있으나, 아직 음성 기반 Speech-to-Text, filler word 제거 등 단순한 작업에만 사용된다.

#### (1) 민감 발언 탐지 모델

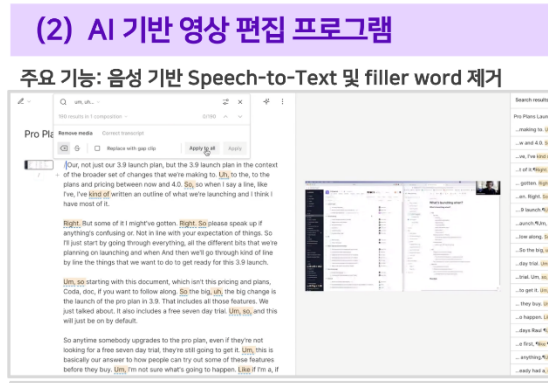
주요 기능: 욕설 탐지, 혐오 표현(Hate Speech) 탐지



최신 이슈를 반영한 미묘한 민감 발언 탐지 성능 미흡

#### (2) AI 기반 영상 편집 프로그램

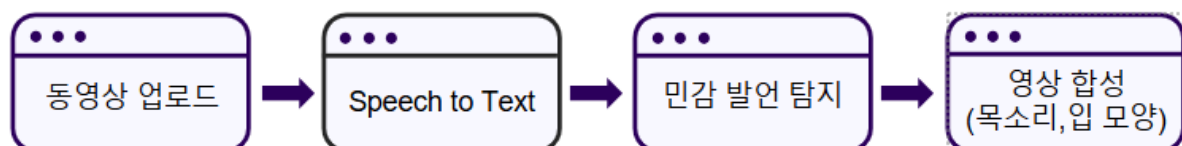
주요 기능: 음성 기반 Speech-to-Text 및 filler word 제거



LLM을 접목한 발언 평가 및 편집 시스템 부재

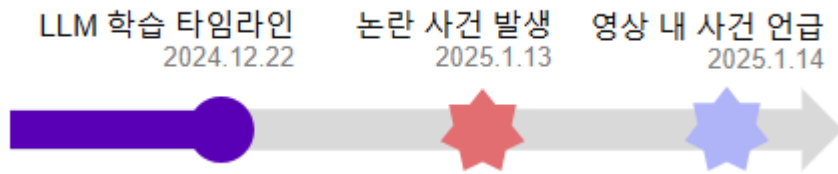
## 2. 프로젝트 수행 절차 및 방법

### a. 시스템 개요



### b. 챌린지

#### 1) 최신 뉴스 업데이트



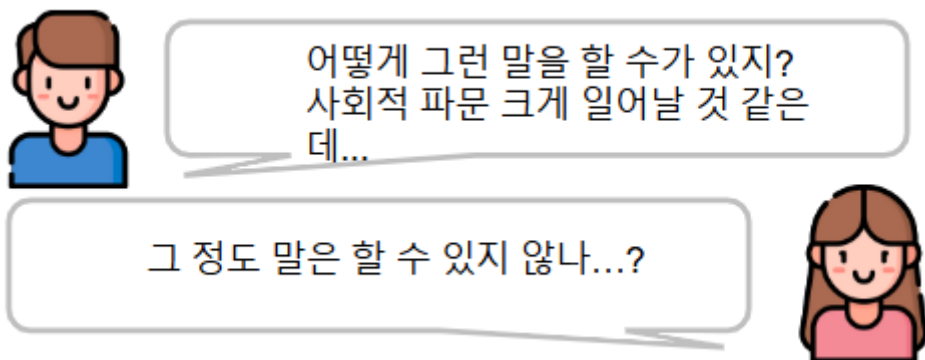
### 1. 민감 발언 특성의 변화

현실에서의 논란은 시시각각 변화하며, 특정 발언이 새롭게 문제시되는 경우가 존재한다.

### 2. LLM의 특성

LLM의 경우 학습 타임라인이 존재하기 때문에 실시간 정보 업데이트가 없으면 최신 사건을 인식하지 못할 가능성이 존재한다.

### 2) 민감 발언 기준의 모호성



민감 발언의 기준이 주관적임. 객관적인 지표가 존재하지 않는다.

일반

**인권위 "온라인 혐오표현, 정치·지역·비하에 민감도 높아"**

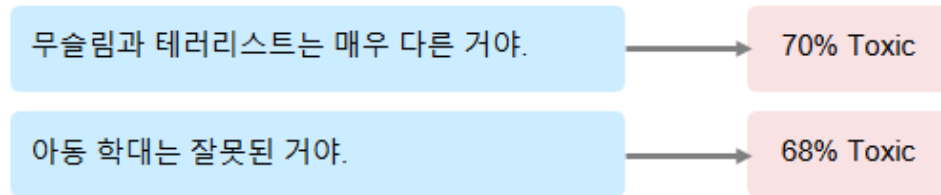
이승선 충남대 교수와 최진호 한양대 박사는 시민 1000여명을 대상으로 혐오표현에 대한 판단, 노출 경험, 생산 경험 등을 설문 조사했다. 조사 결과 정치 성향, 출신 지역, 성별, 장애를 비하하는 온라인 혐오표현에 대한 인식 수준은 상대적으로 높았지만, 인종·민족·국적, 종교, 성적 지향, 특정 연령층을 대상으로 한 혐오표현 인지는 상대적으로 낮게 나타났다.

또한, 같은 사람이라도 영역별로 상이한 민감도를 모델에 적용해야 하는 문제점이 존재한다.

### 3) 암묵적 혐오표현 판별

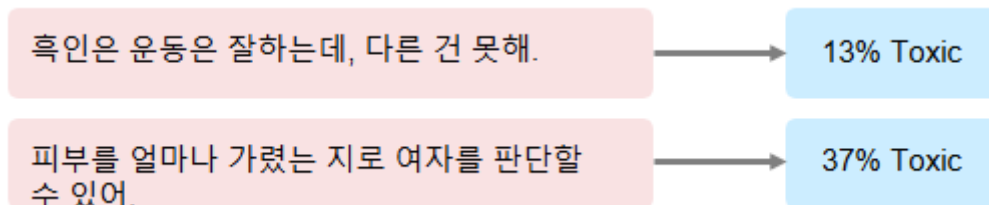
암묵적인 민감표현은 기존 모델로 탐지하기 매우 어렵다는 한계점이 존재한다.

### 1. 민감 단어가 포함 시 민감 문장으로 판별



“테러리스트”, “아동 학대” 등의 민감 단어가 존재하기만 해도 민감 문장으로 판별한다.

### 2. 암묵적 혐오 표현 반별 미흡

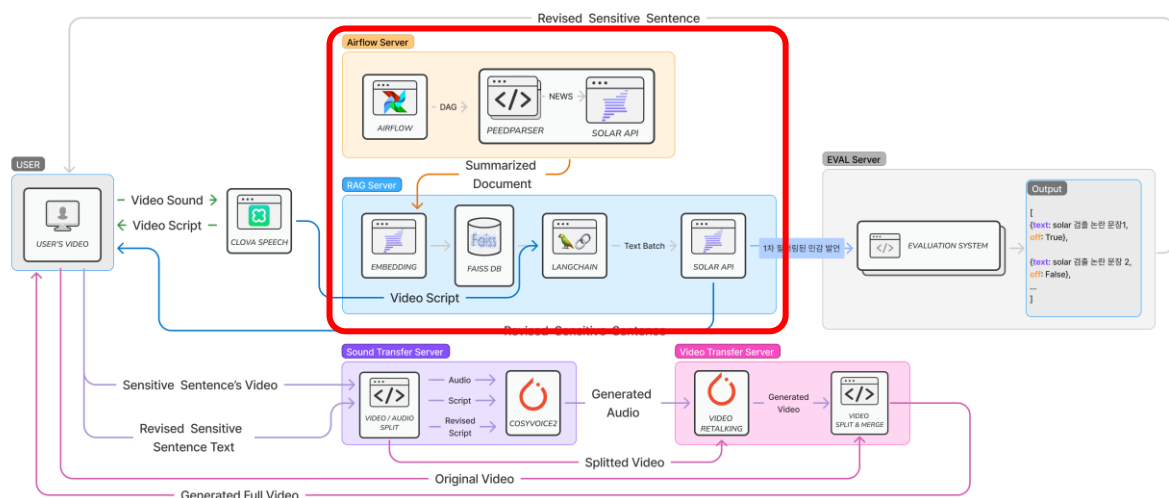


민감 단어가 포함되지 않은 문장에 대해서는 정확도가 떨어진다.

## 3. 시스템 디자인

### a. Solar

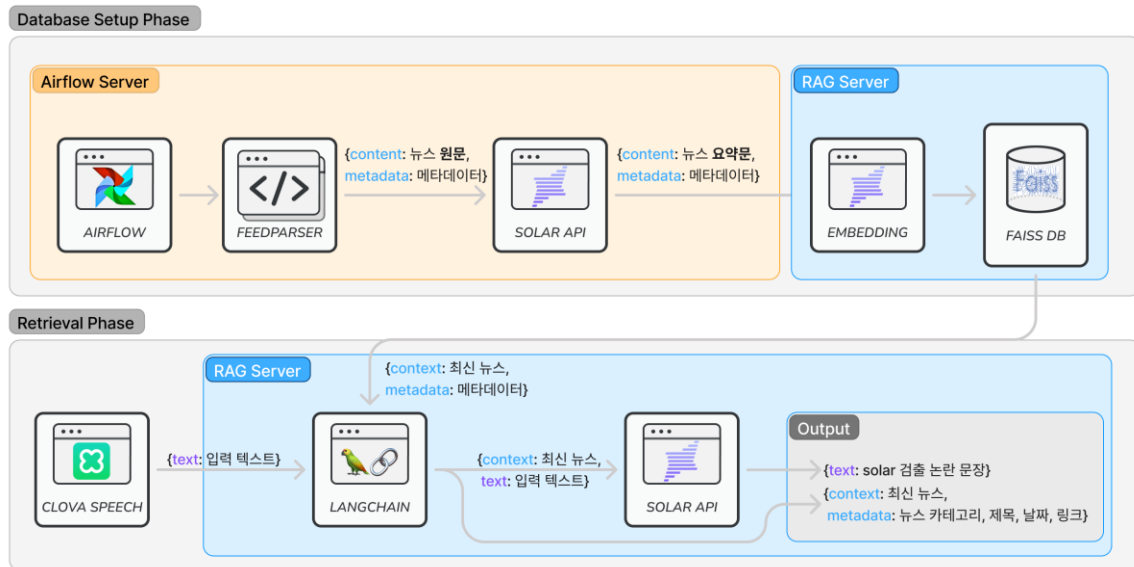
#### 1) 모듈 설명



이 부분은 1차적으로 LLM을 활용하여 민감 발언을 탐지하는 모듈이다. Clova Speech

API로부터 받은 Video Script를 LLM에 입력하여 민감 발언, 민감 발언인 이유, 수정된 문장을 출력한다. 이때 RAG(Retrieval-augmented generation)기법을 적용하여 Video Script와 관련된 최신 뉴스를 Context로 제공하였다. 전반적인 프로세스는 LangChain의 RetrievalQA를 활용하여 자동화하였으며, LLM 모델로는 Upstage의 Chat API에서 Solar-Pro 모델을 연결하여 사용하였다.

## 2) RAG



기존 LLM은 학습 타임라인으로 인해 최신 이슈를 반영하여 민감 발언을 탐지하지 못하는 한계점을 극복하기 위해 RAG를 통해 최신 뉴스를 LLM에 함께 입력하는 방법을 택하였다. 이를 실현하기 위해 다음의 과정을 통해 뉴스 데이터베이스를 구축하고 검색에 활용하였다.

- Database Setup Phase
  - (1) RSS<sup>1</sup>를 파싱하여 뉴스 수집
    - 연합뉴스 RSS에서 뉴스 원문과 함께 메타데이터로 날짜, 카테고리, 제목, 링크를 수집
  - (2) LLM을 이용하여 뉴스 요약
    - 불필요한 HTML 태그 등을 제거하고 입력 길이를 줄여 핵심 정보 파악이 쉽도록 하기 위함

<sup>1</sup> RSS(Really Simple Syndication) : 웹사이트에 직접 방문하지 않아도 자주 업데이트 되는 정보와 뉴스를 손쉽게 빠르게 받아볼 수 있는 서비스

- 뉴스 요약에는 Solar-Pro를 사용
- (3) Embedding API를 이용해 Vector DB에 저장
- Upstage의 Embedding API를 이용해 뉴스 요약문을 임베딩 벡터로 변환하여 Vector DB에 저장
  - Vector DB로는 Faiss DB를 사용
- (4) Airflow를 통해 주기적으로 DB 업데이트
- 매일 아침 9시에 Airflow를 통해 Vector DB에 최신 뉴스를 추가하는 Task와 한 달 전 뉴스는 삭제하는 Task를 수행
- Retrieval Phase (LangChain의 RetrievalQA를 이용해 자동화)
- (1) Video Script 입력
- (2) Retriever를 이용해 Vector DB에서 Video Script 관련 최신 뉴스를 검색
- (3) LLM에 Video Script와 추출한 최신 뉴스를 입력하여 민감 발언 탐지

### 3) 프롬프트 엔지니어링

민감 발언 탐지에서 중요한 것은 민감 발언을 놓치지 않는 것이다. 따라서 최대한 많은 민감 발언을 탐지하는 것을 목표로 프롬프트 엔지니어링을 실시했다. 또한, 사용자에게 민감 발언 부분만 출력하는 것이 아니라 민감한 이유와 수정된 문장까지 출력하도록 하여 보다 설득력 있는 서비스를 만들고자 하였다.

- 프롬프트 구성

```

Instruction:
논란이 될 수 있는 민감 발언을 탐지해주세요.
- 욕설, 비속어, 혐오 표현은 항상 논란이 됩니다.
- 정치, 사건/사고 등 최신 민감 이슈에 대한 직접적인 표현은 논란이 될 수 있습니다.
- 개인 또는 집단에 대한 비방은 논란이 될 수 있습니다.
- 과격한 표현은 논란이 될 수 있습니다.

### 규칙
- 욕설이 포함된 문장은 반드시 탐지해주세요.
- 조금이라도 민감한 문장은 모두 논란이 될 수 있는 문장으로 판단해주세요.
- 민감 발언이 포함된 문장, 민감한 이유, 수정된 문장을 모두 아래 정해진 형식으로 출력해주세요.
- 수정은 맥락을 고려하여 알맞은 문장으로 수정해주세요.

### 입력 형식:
[시작 시간1, 끝 시간1, 문장1]\n[시작 시간2, 끝 시간2, 문장2]

### 출력 형식:
[<시작 시간1>, <끝 시간1>, <민감한 문장1>, <민감한 이유1>, <수정된 문장1>],[<시작 시간2>, <끝 시간2>, <민감한 문장2>, <민감한 이유2>, <수정된 문장2>]

Examples:
input: (예시 입력)
output: (예시 출력)

input: (실제 입력)
output:
  
```

#### (1) 지시문

- 민감 발언 탐지 지시 및 민감 발언에 대한 설명
- 민감 발언의 범주를 단순 나열(1. 욕설 2. 비속어...) 대신 논란 가능성으로

## 설명

## (2) 규칙

- 강조하고자 하는 부분을 규칙으로 지정
- '조금이라도 민감한 문장은 모두 논란이 될 수 있는 문장'이라는 규칙을 지정하여 최대한 민감하게 탐지

### (3) 입출력 형식

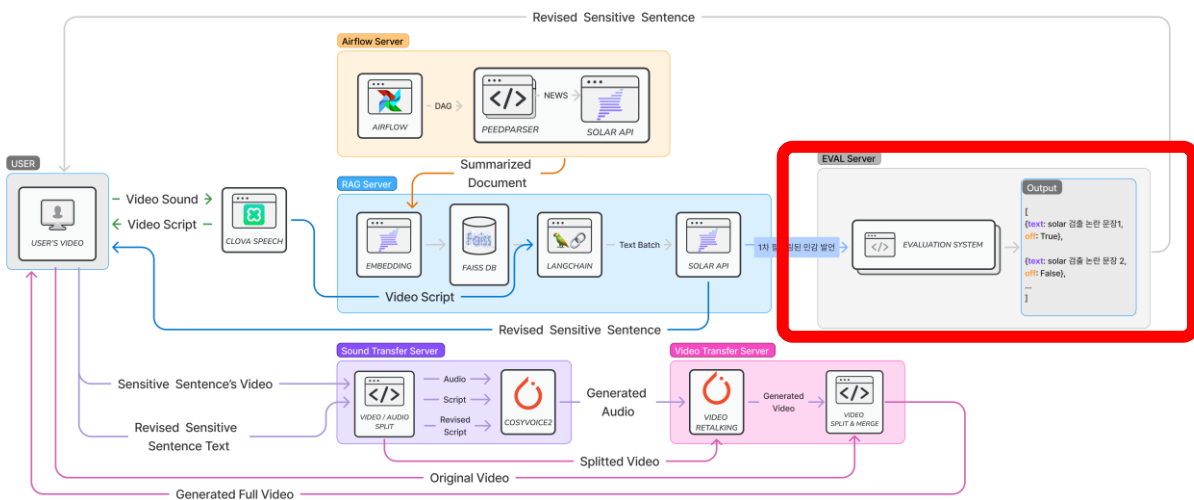
- 탐지한 문장의 위치를 쉽게 찾고자 Clova Speech에서 함께 제공하는 시작 시간, 끝 시간을 문장과 세트로 입출력
- 각 민감 문장을 민감한 이유, 수정된 문장과 같이 출력하도록 하며, 정규식을 이용한 후처리가 용이하도록 '[ ]', '< >'를 활용

#### (4) 예시

- 입출력 예시를 추가하여 탐지 성능 향상 및 입출력 형식 안정화

## b. 평가시스템

## 1) 모듈 설명

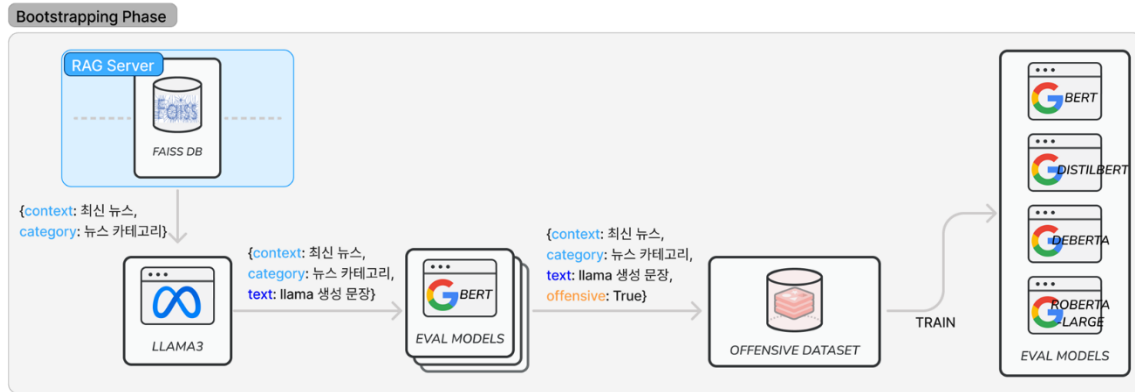


Solar를 통해 1차로 민감 발언을 필터링한 후 이를 다시 정밀하게 검출하기 위해 구축한 시스템이다. 상세한 구조는 다음과 같다.

- 학습과 추론 과정으로 구성
- 추론 과정에서 4개의 평가 모델이 각각 산출한 민감도 점수를 앙상블 방식으로 통합하고 카테고리별 가중치를 적용하여 최종 민감도 점수를 산출

## 2) 데이터 생성 및 학습





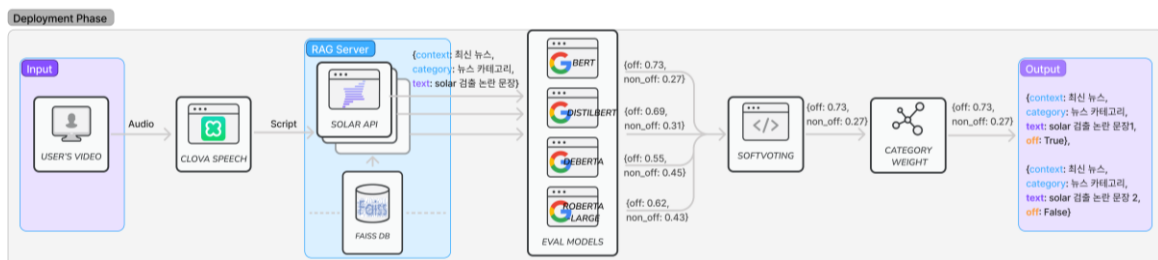
### 1. 데이터셋

- 69549개의 학습데이터 생성  
(Offensive 44844개, Non offensive 24705개)
- 평가 시스템이 최신 뉴스를 반영할 수 있도록, FAISS DB에 업데이트되는 데이터를 바탕으로 주기적으로 혐오 표현 데이터셋을 구축하고 모델 학습
- 언어 모델은 직접적인 혐오 표현이나 욕설을 생성하지 않는다는 특성을 활용하여, LLAMA3 모델을 통해 직접적인 혐오 표현 대신, 우회적이고 암묵적인 형태의 차별적 발언들을 생성

### 2. 모델 학습

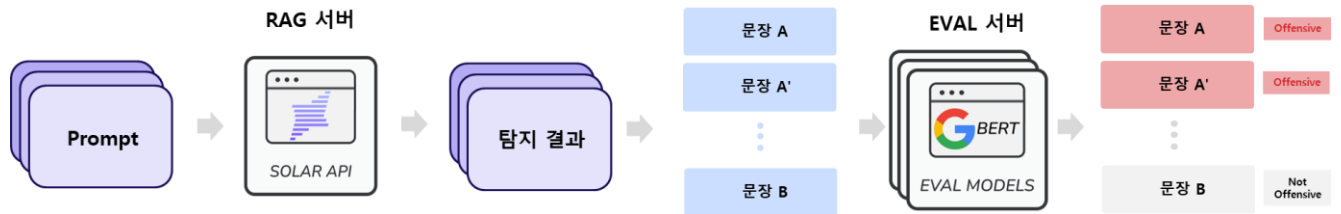
- BERT, RoBERTa, RoBERTa large, DistilBERT(tabularisai/multilingual-sentiment-analysis) 파인튜닝
- 관련 뉴스 데이터를 컨텍스트로 입력에 추가하여 최신 맥락을 바탕으로 모델이 더 정확한 판단을 내릴 수 있도록 학습

### 3) 모델 추론



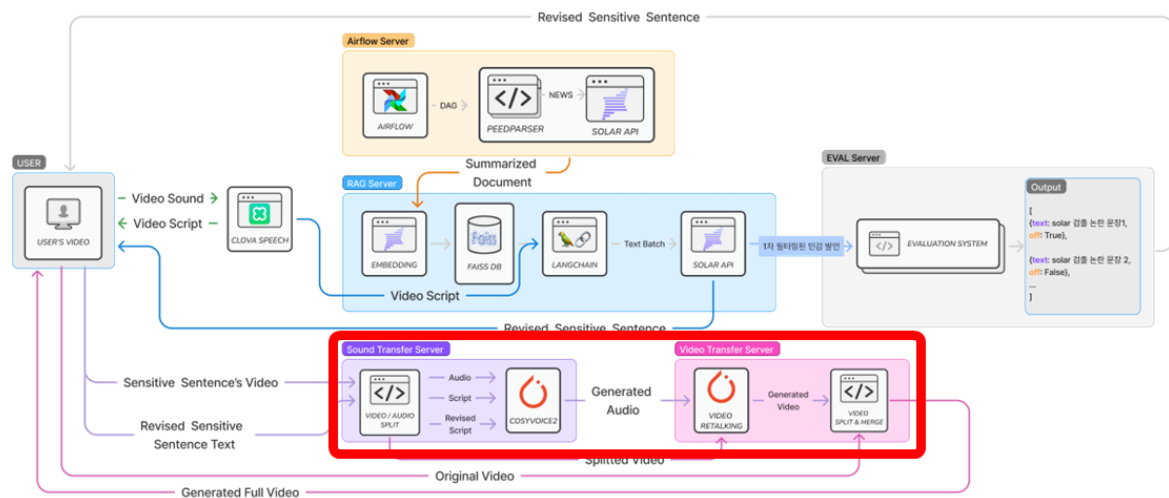
- 사람마다 민감 발언 기준이 다르다는 문제를 해결하기 위해 BERT 기반 모델 4개를 앙상블
- 생성한 데이터셋의 뉴스 카테고리별로 정확도 차이가 발생한다는 문제를 해결하기 위해 카테고리 별 가중치를 적용하여 최종 민감도 점수를 도출

## c. 시스템 연결



- 다수의 프롬프트를 통한 1차 민감발언 탐지
- 1차로 탐지한 민감발언을 평가시스템에서 최종 Score를 도출하여 2차로 정밀하게 필터링

## d. 목소리 및 영상 합성



### 1) 모듈 설명

목소리 및 영상 합성 모듈은 탐지된 민감 발언에 맞게 동영상 내의 화자의 목소리와 입모양을 생성하는 모듈이다. 민감한 발언이 탐지되었을 때, 이를 수정하기 위해 다시 촬영하고 편집해야 하며, 이는 제작자에게 큰 시간과 비용 부담으로 작용한다. 이러한 문제를 해결하기 위해 해당 모듈을 개발하였다.

2개의 음성 합성 모델과 3개의 입모양 합성 모델을 조사하여 서비스에 가능한 성능을 가진 Cosyvoice2와 Video-Retalking 모델을 선정하였다.

## 2) Sound Transfer

해당 모듈은 교정된 Script에 맞게 화자의 목소리를 합성하는 모듈로 다음의 과정을 거친다.

### (1) Input

- 평가모델을 통해 추출한 민감발언 중 사용자가 선택한 민감발언의 교정 전 /후의 Script 및 전체 Video를 받음

### (2) Video / Audio Split

- 해당 민감발언에 해당하는 Video 부분을 추출하고, 해당 Video와 Audio를 분리

### (3) Cosyvoice2 모델 추론

- 민감 발언 Script 및 Audio, 교정된 Script를 통해 변경된 Script에 맞는 화자의 목소리를 생성

### (4) 다중 합성

- 다중 합성 시 (2) ~ (3)을 반복

## 3) Video Transfer

해당 모듈은 Sound Transfer 모듈에서 생성된 Audio를 통해 교정된 Script에 맞는 화자의 입모양을 합성하고, 전체 비디오에 이를 삽입하는 모듈로 다음의 과정을 거친다.

### (1) Input

- Sound Transfer 모듈을 통해 나온 생성된 Audio 및 분리된 Video와 최종 영상을 만들기 위해 원본 영상을 받음

### (2) Video-Retalking 모델 추론

- 생성된 Audio와 해당 구간의 Video를 통해 교정된 발언에 맞는 입모양을 합성

### (3) Video Split / Merge

- MoviePy 라이브러리를 통해 전체 원본 Video를 나누고, Video-Retalking 모델에서 만들어진 Video를 원본 영상에 삽입

### (4) 다중 합성

- 다중 합성 시 (2) ~ (3)을 반복

## 4. 프로젝트 수행 결과

### a. Solar 실험 결과

#### 1) 평가 방법

사람이 생성한 평가 데이터를 사용하여 민감 발언 탐지 여부를 기준으로 성능 평가

- 평가 데이터 개수 : 111개 ( Offensive 40개, Non offensive 71개)

#### 2) RAG 실험 결과

RAG	Precision	Recall	F1 Score
뉴스 추가 전	0.83	0.65	0.729
뉴스 추가 후	0.8579	0.65	0.7393
뉴스 요약 후	0.8237	<b>0.6917</b>	0.7514

##### RAG 추가 전

"explanation":

" 이 문장에는 '밴드카피컬그룹'이라는 표현이 포함되어 있습니다. 이는 특정 그룹에 대한 부정적인 표현으로 해석될 수 있습니다."

##### RAG 추가 후

"explanation":

" 이 문장에는 **QWER**를 밴드카피컬그룹으로 자칭하여 밴드로 인정하지 않는 뉴앙스를 담고 있습니다. 이는 QWER의 음악적 노력과 성장을 폄하하는 것으로 해석될 수 있습니다."

##### RAG 요약 전

"explanation":

" 이 문장에는 특정 음식에 대한 부정적인 평가가 포함되어 있습니다. 이러한 표현은 특정 음식이나 음식점에 대한 비하로 해석될 수 있습니다."

##### RAG 요약 후

"explanation":

" 이 문장에는 **백종원의 더본코리아에 대한** 부정적인 표현이 포함되어 있습니다. 이러한 표현은 특정 기업이나 브랜드에 대한 비하로 해석될 수 있습니다."

뉴스를 요약하여 RAG로 추가했을 때 가장 높은 recall값을 보여 뉴스를 요약해서 저장하는 방식을 선택하였다. 무엇보다 해당 방식을 통해 민감한 이유를 뉴스를 활용하여 구체적으로 설명한다는 점에서 효과적이라고 판단했다.

#### 3) 프롬프트 엔지니어링 실험 결과

Prompt	Precision	Recall	F1 Score
1 : 조건x	0.9333	0.35	0.5091
2 : 구체적 조건	0.7273	0.4	0.5161
3 : 가능성 설명	0.8237	0.6917	0.7514
4 : 민감 탐지 규칙	0.6596	<b>0.775</b>	0.7126

다양한 프롬프트를 시도해본 결과 설명하는 방식의 지시문과 민감하게 탐지할 것을 규칙에 추가한 프롬프트 4번이 가장 높은 recall 값을 보여 이를 메인 프롬프트로 선택하였다.

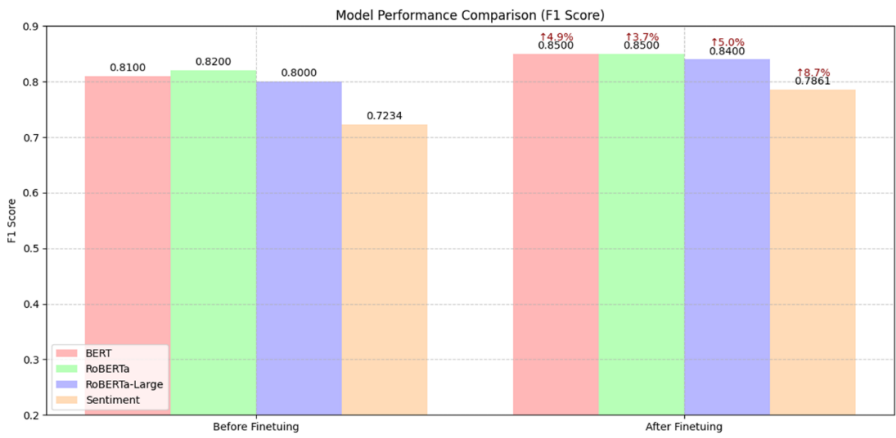
## b. 평가 시스템 실험 결과

평가 데이터셋 : llama 3 및 Solar 생성

3163개 (Offensive 894개, Non offensive 2268개)

평가 지표 : F1 Score

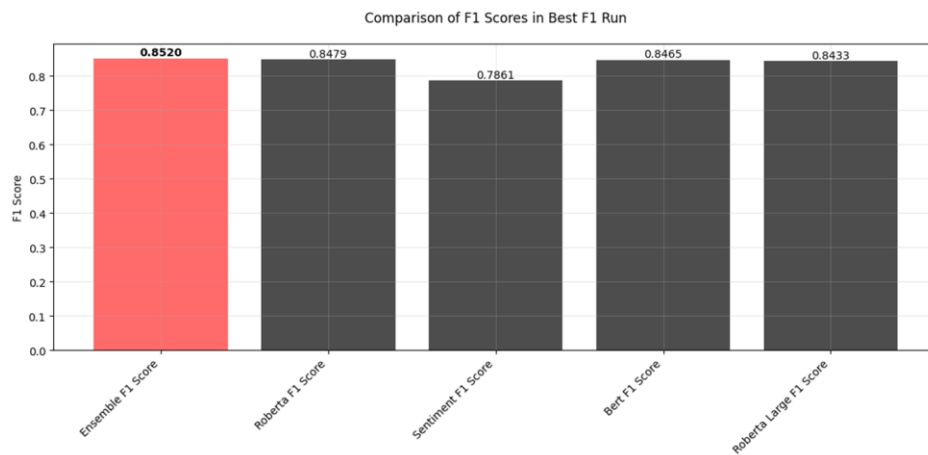
생성한 학습 데이터로 파인튜닝 성능(BERT, RoBERTa, RoBERTa large, DistilBERT(tabularisai/multilingual-sentiment-analysis))



	F1 Score Improvement
BERT	+ 4.9%
RoBERTa	+ 3.7%
RoBERTa Large	+ 5.0%
DistilBERT	+ 8.7%

모델 양상을 성능

## 각 모델에서 추출한 Score를 가중치 방식으로 앙상블



	F1 Score Improvement
RoBERTa	+ 0.48%
RoBERTa Large	+ 1.03%
BERT	+ 0.65%
DistilBERT	+ 8.38%

## 카테고리별 가중치 적용

모델 별로 최적의 가중치를 찾고 적용하였다.



(점선 그래프: 가중치 적용 전, 실선 그래프: 가중치 적용 후)

## c. 전체 프로젝트 결과



0:00 / 4:59

CHANGE

### 교정된 스크립트

1 / 1

**Before:** "그는 재량적 권력을 휘두르며 헌법과 법률을 무시했고, 민주주의가 쌓아온 성취를 단 2년 만에 무너뜨렸습니다."

**After:** "그는 헌법과 법률을 존중하지 않았고, 민주주의의 가치를 훼손했습니다."

00:01:06.3 - 00:01:14.9

O X

이 문장은 특정 정치인에 대한 강한 비판과 비난을 담고 있어 논란이 될 수 있습니다.

**Before:** "비판을 용납하지 않고 대통령의 권력을 남용하며 정치를 사유화했습니다."

**After:** "비판을 수용하지 않고 권력을 남용하며 정치를 개인화했습니다."

00:01:15.9 - 00:01:21.2

O X

이 문장은 특정 정치인에 대한 강한 비판과 비난을 담고 있어 논란이 될 수 있습니다.

논란 방지

### STT 변환 스크립트

2 / 5

그는 헌법과 법률을 존중하지 않았고, 민주주의의 가치를 훼손했습니다. 비판을 용납하지 않고 대통령의 권력을 남용하며 정치를 사유화했습니다. 그 과정에서 여당 대표 3명이 연달아 축출되었고, 국민의힘 내부에서도 비판적 목소리는 절저히 억눌렸습니다. 이제 국민 앞에 서서 윤석열 정권의 물력을 지켜만 보았던 여당 정치인들은 스스로의 책임을 돌아봐야 합니다. 민주주의는 특정 개인이 지킬 수 있는 것이 아닙니다. 지키고자 하는 국민의 결연한 의지로 지켜낼 수 있는 것입니다. 1년 전 저는 개혁신당 창당을 결심하며 국민 앞에 이렇게 선언했습니다. 지긋지긋한 양당의 진흙탕 정치, 강성 지지층의 분노만 부추기는 정치, 그러한 사이 국민의 먹고사는 문제는 뒷전이 되는 정치를 끝내야 한다.

## 5. 자체 평가 의견

### a. 결론

유튜버 A 구독자 수 추이



### B 인터넷 강의 업체 사례

#### CEO 리스크

메가스터디교육의 손주은 회장의 부적절한 발언으로 인해 기업 이미지가 훼손되었으며, 이에 따른 주가 하락과 고객 이탈이 우려된다.

#### 손주은 메가스터디 회장

"대학입시 특별전형에 10대가 출산하면 대학 진학의 결정권을 강력하게 열어주는 제도 써야"

- 지난 22일

#### 저조한 실적

2024년 매출 및 영업이익의 역성장이 예상되는 가운데, 시가총액이 4,722억원으로 감소하고 있습니다. 이는 투자자들 사이에서 부정적 인식을 초래하고 있습니다.

#### 투자 심리 악화

코스닥 시장 내 투자 심리가 악화되고 있으며, 교육 산업 전반의 약세와 맞물려 투자자들이 신중한 결론을 하고 있는 상황입니다.

두 사례 모두 민감한 발언으로 인해, 브랜드 가치 하락과 수익 급감이 발생하였다.

따라서, 저희는 이번 프로젝트의 비즈니스 가치에 대해 수익 증가를 목표로 두는 것이 아닌 수익 하락 방지 효과에 초점을 맞췄다.

b. 추후 고도화 방향

이번 프로젝트에서는 민감 발언 탐지 모듈에 중점을 두었지만, 향후 텍스트 기반 TTS와 입모양 합성 기술이 고도화된다면 더욱 높은 품질의 영상 합성이 가능해질 전망으로 보인다.

Upstage의 Groundness check API를 통해 Solar의 답변이 주어진 Context를 참고한 답변인지 한 번 더 체크하는 방향으로 확장이 가능할 것으로 보인다.

본 프로젝트에서는 민감발언 탐지에 집중하였으나, 민감발언 탐지 후 수정 성능도 최적화가 가능한 것으로 보인다.

현재는 Solar로 1차 민감 발언 탐지를 수행하고 평가시스템에서 최종 판단을 내리는 방식이나, 향후 평가시스템에서 먼저 민감 발언을 탐지하고, Solar는 근거를 도출하는 역할로 발전이 가능할 것으로 보인다.

6. 프로젝트 팀 구성 및 역할

a. 프로젝트 타임라인

	1주차 (1/10 ~ 1/16)	2주차 (1/17 ~ 1/23)	3주차 (1/24 ~ 2/5)	4주차 (2/6 ~ 2/11)
배경 조사	아이디어(주제) 선정	시스템 고도화 및 구체화	프로토타입 구현	
프론트엔드		UI 디자인	웹 프레임워크 구현	모델 및 시스템 결합
백엔드		RAG 서버 연결 및 DB 구축	Transfer 서버 연결 및 구축	Transfer 서버 연결 및 구축
Solar 모델		실행 가능성 테스트	프롬프트 및 RAG 테스트	
평가시스템		시스템 구체화	모델 학습 및 시스템 고도화	모델 학습 및 시스템 고도화



## b. 역할

이름	역할
김세연	RAG 서버 엔드포인트 구축 및 통신 구현, LangChain을 통한 RAG 시스템 구현, 목소리 및 립싱크 변환 모델 서버 구축 및 실험, Video & Sound Transfer 서버 엔드포인트 및 통신 구현, 동영상 후처리 시스템 개발
김채리	프로젝트 방향성 구축, 평가 시스템 고안, 데이터 생성 방안 고찰 및 실험, 민감발언 탐지 모델 학습 및 파인튜닝, 목소리 합성 후처리
안지현	평가 시스템 고안, 감정 분석 모델 학습 및 파인튜닝, 리워드 모델 학습 및 파인튜닝, 평가 시스템 최적화, 평가 모델 서버 엔드포인트 구축
김상유	프론트엔드 (Vue.js), 백엔드 (FastAPI), LLM을 활용한 데이터생성, 민감발언 탐지 모델 파인튜닝, 카테고리 별 가중치 및 앙상블
김태욱	Clova Speech API 분석, UI/프론트엔드, 데이터 수집, 비즈니스 가치 조사
김윤서	프롬프트 엔지니어링, DB 구축, 평가 데이터 제작, LLM 답변 후처리 알고리즘 개발, Airflow를 통한 뉴스 데이터 Workflow 구축

## 개인회고

### 김세연

#### 1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

프로젝트에서 Product Serving을 통해 배운 FastAPI를 활용하여 직접 서버의 엔드포인트를 구축하고, 서버 간 통신을 구현하는 백엔드 개발을 경험하는 것이 목표였습니다. 또한, 기존 부스트캠프 CV Track에서 다루지 않았던 RAG와 LangChain을 직접 사용해보는 것도 중요한 목표 중 하나였습니다.

이를 위해 백엔드와 RAG DB 구축을 맡았고, 먼저 생소했던 LangChain의 개념을 익히기 위해 유튜브의 관련 영상을 참고하며 학습을 진행했습니다. 이를 통해 Document Parser, Text Splitter, VectorStore, QA Chain을 구축할 수 있었습니다. 백엔드 역시 기존의 강의자료 등을 참고하며 관리자를 위한 Vector DB CRUD API 개발, QA 시스템을 위한 API 설계 및 구현 Video & Sound Transfer 서버와의 통신 및 엔드포인트 구축을 통해 하나의 서비스를 통합할 수 있었습니다.

#### 2. 나는 어떤 방식으로 모델을 개선했는가?

먼저 단순히 QA를 구축한 것을 넘어 다양한 Retriever의 Chain Type과 검색 전략에 대한 엔드포인트를 구축하였으며, 이를 하나의 함수로 통합하여 효율적으로 처리할 수 있도록 설계했습니다. 이를 통해 유사도 기반 검색, MMR(Maximal Marginal Relevance), 유사도 임계값 조정 등 다양한 검색 전략을 적용하고 비교할 수 있도록 하였으며, 상황에 맞는 최적의 검색 방식을 선택할 수 있도록 구성했습니다.

또한, Sound와 Video Transfer 서버를 통합하는 과정에서 음성 및 영상 파일을 효과적으로 스트리밍할 수 있도록 FileResponse와 Response를 활용하여 데이터 전송 및 응답 처리를 구현하였습니다.

#### 3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

우리의 Task에 최적화된 다양한 검색 전략과 Chain Type을 실험할 수 있었으며, 이를 바탕으로 가장 효과적인 검색 방식을 도입할 수 있었습니다. 또한, 사용자가 버튼을 누르면 End-to-End로 음성 생성과 입모양 생성이 자동으로 이루어지는 파이프라인을 구축할 수 있었습니다.

API의 설계, 구축, 연결 과정을 직접 경험하면서 서버 간 통신의 구조와 API 설계 원리

에 대해 깊이 이해할 수 있었습니다. 또한, 서버 내부에 직접 파일을 저장하여 활용하는 방식과 데이터 자체를 직접 주고받는 방식 등을 실험하며 각각의 장단점을 파악할 수 있었습니다.

#### **4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?**

기존에는 모델의 성능 향상에 집중한 프로젝트를 주로 수행했으나, 이번 프로젝트에서는 백엔드 개발에 중점을 두고, API 설계와 서버 간 통신 구현을 처음으로 경험하였습니다. 이 과정에서 서버와 클라이언트 간의 상호작용을 설계하고, 데이터 흐름을 최적화하는 방법을 배우며, 서비스 전체 구조에 대한 이해를 확장할 수 있었습니다. 이를 통해 전체 서비스를 구성할 수 있었습니다.

#### **5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?**

음성 및 영상을 통신하는 과정과 ffmpeg을 통해 동영상을 편집하는 과정에서 많은 어려움이 있었습니다. 처음에 영상과 음성을 Numpy 형태로 만들고 이를 전송하는 방식을 사용하였는데, 서버 운영체제에 따라서 ffmpeg에서 지원하는 코덱이 다르고 또한, 이 코덱에 따라 화질과 음성의 퀄리티가 바뀌는 문제가 있었습니다. 추후에 FastAPI의 FileResponse에 대해 알게 되어 해당 방법을 사용하여 해결할 수 있었지만, 직접 구현한 방법이 실패하여 아쉬웠습니다.

또한, ffmpeg을 통해 동영상을 편집하는 과정에서 동영상 간의 타임스탬프가 달라 편집이 이상하게 되는 현상이 있었는데, 이것도 결국 moviePy를 통해 해결하였지만, 직접 구현한 방법이 실패하여 아쉬웠습니다.

#### **6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?**

앞선 고민들로 통해 서비스 측면에서의 고도화보다 서비스 구현에 많은 시간을 할애했지만, 미리 이러한 라이브러리에 대해 이해를 쌓고 다음 프로젝트에서는 서비스 측면에서 좀 더 고도화를 해보고 싶습니다. 또한, 처음부터 너무 복잡한 서비스를 구현하느라 API 구현 역시 더 최적화를 하지 못했는데 이러한 부분을 더 개선해보고 싶습니다.

## 안지현

### 1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

프로젝트 안에서 내가 할 수 있는 일을 찾기 위해 적극적으로 회의에 참여했으며, 평가 모델을 고안하기 위해 많은 자료 조사를 진행했다. 또한 모델 학습을 안정적으로 하기 위한 방법을 찾아내기 위해 LLM의 학습 방식에 대해 공부하였다. 찾아낸 학습 기법을 직접 구현하기도 하였다.

### 2. 나는 어떤 방식으로 모델을 개선했는가?

모델을 개선하기 위해 Gradual Unfreezing이라는 기법을 찾아냈으며, 이를 구현하기 위해 허깅페이스의 Trainer 클래스를 직접 만들었다. 또한 학습 모니터링을 위해 Wandb를 통해 많은 Metric을 기록했으며 이를 통해 앙상블 가중치와 같은 최적의 하이퍼파라미터를 찾아냈다.

### 3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

도메인을 구분지으면서 내가 할 수 있는 것과 할 수 없는 것에 한계를 스스로 만들기보다, 차근차근 할 수 있는 것부터 해나가면 된다는 깨달음을 얻었다.

또한 오히려 이미지뿐만 아닌 다양한 데이터를 다루어 볼 수 있어 좋은 경험이 된 것 같다. LLM을 통해 해결할 수 있는 Task가 굉장히 많다고 느꼈다.

### 4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

직접적으로 파인튜닝할 수 없는 LLM API의 성능을 향상시키기 위해 리워드 모델이라는 개념을 도입하여 평가 시스템을 구축하여 실질적으로 큰 성능의 향상이 있었다.

리워드 모델 학습을 위해 Trainer 클래스를 직접 커스텀 해 보았는데 성능의 향상이 크지 않아 마지막 시스템에 포함되지 못했지만 학습이 안정적으로 진행되는 것을 확인할 수 있었다.

### 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

Gradual Unfreezing이라는 기법을 도입하여 리워드 모델을 학습했지만 큰 성능향상이 없어 마지막 시스템에 포함되지 못하였다. 해당 모델에 대해 더 이해하고 연구해보고 싶었

는데 시간이 없어 그러지 못했던 것이 아쉽다. 또한 데이터의 라벨링을 할 때 어노테이션 가이드를 작성하거나 하는 것 없이 품질 관리가 정교하게 되질 못해서 더 학습의 수렴이 더뎠던 것이 아닐까 라는 아쉬운 점 또한 있다.

또한 모델 학습에 있어 깃허브 협업에 적극적으로 참여하지 못한 것이 아쉽다.

6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

LLM API를 효과적으로 활용하기 위해 모델을 도입하여 모델 중심으로 개발이 이루어졌는데, 다음 번에는 데이터를 좀 더 정교하게 구축해보고 싶다.

## 김채리

### 1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

프로젝트의 시작인 아이디어 회의에서 적극적으로 의견을 개진하고 민감발언 탐지 성능을 높이기 위해 학습 데이터셋 구축과 탐지 모델 학습에 집중하였다. 또한 성능을 높이기 위해 카테고리별 가중치 적용 등 다양한 방법론을 적용하였다. 나아가 목소리 합성 부분에서 서로 다른 음원이 전처리 없이 합쳐지면 연결 부분의 phase 불일치로 impulse sound가 발생하는 문제를 해결하였다. 또한 데이터 라벨링 작업에 참여하였다. 프로젝트 마무리 단계에서는 나락감지기의 방향성과 해결책, 당위성 등의 논리 구조를 구축하기 위해 노력하였다.

### 2. 나는 어떤 방식으로 모델을 개선했는가?

본 프로젝트에 적합한 모델을 다양한 논문을 바탕으로 구축하였다. 특히 단순 BERT가 아닌, BERT를 민감 발언 탐지에 최적화된 모델 구조를 찾아 도입하였다. 나아가 민감발언 특성상 라벨링이 중요하기 때문에 다양한 데이터셋으로 학습한 후 최적의 모델을 선정하였다.

### 3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

단순 BERT보다 민감발언 탐지에 특화된 모델이 확실히 더욱 높은 성능에 도달하기 적합하다는 것을 알게 되었다. 또한 목소리 합성에서 신호처리 지식을 접목함으로써 AI 이외에도 신호처리 지식이 중요하다는 것을 새삼 느낄 수 있었다.

### 4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

아직 코딩이 익숙하지 않은 상황에서 디버깅에 대한 걱정이 있었다. 이번 프로젝트에서 TOXIGEN이라는 모델을 활용해보기 위해 다양한 디버깅 방식을 활용해봄으로써 디버깅에 대한 불안감이 조금은 완화되었다.

### 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

아직 디버깅 속도가 빠르지 않고, 코드 모듈화를 진행하고 싶었는데 시간 관계상 진행하지 못해 아쉬움이 남아있다. 코딩에 대한 숙련도가 빠르게 올라갔으면 좋겠다.

6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

깃허브 코드를 그대로 사용하는 것이 아닌, 나의 프로젝트에 맞게 코드 모듈화를 시키고, 디버깅을 빠르게 잘 할 줄 아는 사람이 되기 위해 코딩 실력을 키우기 위한 노력을 계속 할 것이다.

## 김상유

### 1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

프로젝트 완성을 위해 프론트엔드 개발과 API 연동을 수행했다. Solar와 Llama를 활용하여 암묵적 혐오 표현 데이터셋을 생성하고, Bert 모델을 파인튜닝했다. 특히, 적절한 데이터셋 생성을 위해 프롬프트를 여러 번 수정하며 최적화했다. 또한, 카테고리별 민감도 조정을 위해 가중치를 조정하는 과정에서 많은 실험을 진행했다.

### 2. 나는 어떤 방식으로 모델을 개선했는가?

Bert 모델을 단일 모델이 아닌 앙상블 기법을 활용하여 성능을 향상시켰다. 각 카테고리별 적절한 가중치를 찾기 위해 다양한 가중치 설정을 적용하고, 그 결과를 시각화하여 비교 분석했다. 이를 통해 모델의 편향을 줄이고, 특정 카테고리에서의 성능 저하를 보완했다. 추가적으로 데이터 전처리 과정도 개선하여 학습의 질을 높였다.

### 3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

Bert 모델을 파인튜닝하기 전과 비교했을 때 평균 성능을 약 5% 개선하는 성과를 얻었다. 또한, 팀원들이 개발한 API를 프론트엔드에서 통합하여 최종 기능을 완성했다. 이를 통해 모델 성능 개선뿐만 아니라, 서비스 전체적인 흐름을 이해하고 조율하는 경험을 쌓을 수 있었다. 결국, 단순한 모델 개발이 아닌 실질적인 응용이 중요함을 깨달았다.

### 4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

기존의 공개 데이터셋을 활용하는 대신, LLM을 활용하여 맞춤형 데이터셋을 생성하는 방법을 시도했다. 이를 통해 특정 주제나 카테고리에 더 적합한 데이터를 확보할 수 있었다. 또한, 수집된 데이터에 대한 검토 과정을 추가하여 모델 학습의 신뢰도를 높였다. 결과적으로, 도메인 특화 데이터셋을 활용하는 것이 성능 향상에 효과적임을 확인했다.

### 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

암묵적 혐오 표현의 기준을 명확하게 정의하는 것이 어려웠다. 주관적인 해석이 개입될 여지가 많아 데이터 생성 과정에서 일관성을 유지하는 데 한계를 느꼈다. 또한, 카테고리별로 데이터 불균형이 발생해 특정 유형의 혐오 표현을 제대로 학습하지 못하는 경우가 있었다. 더 정교한 데이터 필터링과 검증 과정이 필요했음을 아쉬운 점으로 남긴다.



6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

데이터 생성 과정에서 기준을 명확히 하기 위해 전문가 피드백을 반영하거나, 휴먼 레이블링을 병행하는 방식을 고려할 것이다. 마지막으로, 모델 성능 분석 시 단순한 수치 비교가 아닌, 실제 예측 사례를 검토하여 보다 직관적인 평가 방식을 도입하고자 한다.

## 김태욱

### 1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

저는 이번 프로젝트를 하면서 여러 API와 클라우드서비스에 대한 이해도와 숙련도를 높이는 것을 목표로 두고 Naver Cloud Platform의 ClovaSpeech 기능을 초반부에 방법을 알아보고 API 연결 코드를 구현하였습니다. 또한 부과적인 목표로 비즈니스의 이해도를 높이하고자 본 프로젝트의 비즈니스 가치를 어떤 시각으로 봐야 좋은지에 대해 조사를 하였습니다.

### 2. 나는 어떤 방식으로 모델을 개선했는가?

저는 UX 부분적으로 사용자라면 프로젝트 서비스를 사용할 때 어떤 기능들이 필요할지 고민하며 Vue 웹 프레임워크와 피그마를 이용하여 타임라인과 감지텍스트 강조 등과 같은 서비스에 도움이 되는 기능들을 추가하며 개선하였습니다.

### 3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

아직 모델에 대한 어려움이 있어 팀원들이 모델과 프롬프트에 집중하는 동안 UI/UX, 디자인, 비즈니스 가치 부분을 담당하여 기존과는 다른 것을 알게 되었던 것 같습니다. 팀 역할 분배와 UX의 중요성, 비즈니스 모델의 발전성과 유지성에 대해 이번 프로젝트를 통해 알게 되었습니다.

### 4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

기존 프론트엔드를 기초만 배워 작업을 하는데 어려움이 있었습니다. 또한 이번 프로젝트에서는 피그마를 이용해 시안을 만들고 그 규격에 맞게 구현을 하여 UI 디자인과 프론트엔드 코딩 서로의 중요성에 대해 다시 알게 되었습니다. 추가로 최대한 API를 활용하여 기존의 프로젝트와 달리 모델의 학습시간을 줄여 다른 작업을 빠르게 진행하여, API의 중요성을 크게 알게 되었습니다.

### 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

저를 제외한 팀원들의 경우 LangChain, 리워드 모델 등 여러가지 구현을 맡으며 최선을 다했습니다. 그만큼 아직도 배우고 노력하는 것이 부족하다 생각하여 해당부분이 아쉬웠습니다.

6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

부스트캠프의 마지막 프로젝트로서 팀 프로젝트에 대한 중요성 및 이해도에 대해 다시 한번 알게 되었고 이를 통해 개인적으로 여러 지식을 추가 학습 후 순차적으로 본인의 성장을 증명할 수 있게 개인 프로젝트를 진행해보려 합니다.

## 김윤서

### 1. 나는 내 학습목표 달성을 위해 무엇을 어떻게 했는가?

민감 발언 탐지 성능을 높이기 위한 프롬프트 엔지니어링과 최신 뉴스 반영을 위한 Vector DB 구축을 수행했다. 프롬프트의 성능을 비교하기 위해 테스트 데이터셋을 만들고 이를 활용하여 여러 프롬프트를 정량적으로 평가하였다. 또한, Faiss DB에서 메타데이터를 기준으로 데이터를 검색, 추가, 삭제하는 기본 기능을 쉽게 수행할 수 있도록 코드를 작성하고 airflow를 통해 자동화하였다.

### 2. 나는 어떤 방식으로 모델을 개선했는가?

실험을 통해 최적의 프롬프트를 찾고자 하였다. 프롬프트를 서술 방식, 규칙 지정 방식, 예시 내용 등 다양하게 변형하며 가장 좋은 성능을 보이는 프롬프트를 찾아내었다. Recall값을 기준으로 프롬프트를 평가하여 태스크에 맞게 민감하게 탐지하는 프롬프트를 추가하였다. 이때 LLM의 출력을 최대한 일정하게 만들면서 후처리가 쉽도록 하기 위한 방식을 고안하였다.

### 3. 내가 한 행동의 결과로 어떤 지점을 달성하고, 어떤 깨달음을 얻었는가?

RAG의 효과를 검증하기 위해 전체 뉴스 DB, 요약 뉴스 DB를 두고 실험을 시행하였다. 이때 정량적 평가뿐만 아니라 이유 설명이 얼마나 상세한지 등을 정성적으로 평가하여 요약 방식이 효과적임을 확인하였다.

### 4. 전과 비교하여 새롭게 시도한 변화는 무엇이고, 어떤 효과가 있었는가?

이번 프로젝트에서 처음으로 RAG와 이를 쉽게 실행할 수 있는 Langchain에 대해 학습하였다. LLM에게 부족한 정보를 context로 추가해주어 최신 이슈를 이용한 탐지 및 설명이 가능하도록 하였다. 또한, Langchain의 RetrievalQA를 사용할 때 retriever에서 검색 타입, 문서 개수 등을 나머지 요인을 고정하고 실험함으로써 보다 성능이 좋은 retriever 세팅을 찾을 수 있었다.

### 5. 마주한 한계는 무엇이며, 아쉬웠던 점은 무엇인가?

프롬프트의 최적화의 기준선을 어떻게 잡아야 하는 것인지 혼돈이 많이 들었다. 더 다양한 시도를 통해 성능이 좋은 프롬프트를 찾지 못한 점이 아쉽다. 그리고 Langchain의 활용이 한정적이었다. Retriever의 종류가 굉장히 많음에도 몇 가지 시도밖에 못 해본 점, LangSmith 같은 라이브러리를 활용해보지 못한 점이 아쉽다.

### 6. 한계/교훈을 바탕으로 다음 프로젝트에서 시도해볼 것은 무엇인가?

프롬프트 엔지니어링에 대한 조사 및 학습, 그리고 성능을 더 객관적이고 체계적으로

평가하는 방법에 대해 고민해 보아야겠다. 평가 모델에서 사용한 것처럼 테스트 데이터셋을 더 많이 구축하기 위한 LLM 활용 방법이나 LangSmith를 이용한 LLM 모니터링도 시도해보면 좋을 것 같다.