2025 한양대학교

SW/ICT/AI 종합학술대회



[분야 : 404 Session Not Found]

제목: 문맥 정보 기반의 한국어 유해 발언 탐지 모델 연구

문지원 (23학번)

2025. 11. 28

한양대학교 ERICA 소프트웨어융합대학 / 한양대학교 ERICA SW중심대학사업단

연구 배경 및 문제 정의

1. 키워드 필터링(Keyword Filter)의 한계

단순 단어 매칭 방식은 문맥을 파악하지 못해 오탐/미탐 발생

"너 진짜 바보처럼 착하다" 오탐(차단됨)

"머리에 든 게 하나도 없네" 미탐(통과됨)

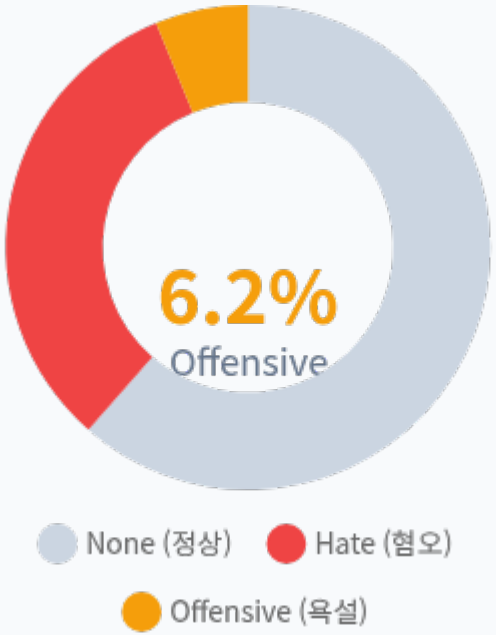
2. 데이터 불균형 (Class Imbalance)

현실 데이터의 롱테일(Long-tail) 분포로 인한 학습 난이도 상승

- 대다수 데이터는 정상(None)에 편중됨
- 모델이 다수 클래스로 편향되어 소수 클래스 탐지 실패

> 핵심 난제 (Challenge)
>
> 가장 탐지하기 어려운 "은근한 욕설(Offensive)"이 데이터셋의 단 6.2%에 불과함

데이터셋 클래스 분포 (총 80,359건)



Offensive 클래스는 데이터가 적어 학습이 매우 어려움

데이터셋 구축 전략

1. 데이터 파이프라인 (Data Pipeline)

Korean HateSpeech
7.9k

UnSmile Dataset
19.2k

KALM Custom
53.3k

→

전처리 (Preprocessing)
중복 제거 / 결측치 제거
특수문자 정제

→

80,359
고품질 학습 데이터 확보

2. 레이블 표준화 (3-Class Labeling)

CLASS 0

None (정상)

욕설이나 혐오 표현이 전혀 포함되지 않은 일반적인 대화 문장

"오늘 날씨 정말 좋네요, 산책 가고 싶다."

CLASS 1

Offensive (욕설)

특정 대상을 지칭하지 않거나, 불쾌감을 줄 수 있는 일반적인 욕설

"아 진짜 이 새끼 또 시작이네 짜증나게."

CLASS 2

Hate (혐오)

성별, 출신, 장애 등 특정 집단을 비하하거나 조롱하는 혐오 발언

"OOO들은 다 거짓말쟁이야, 믿을 게 못 돼."

모델링 및 최적화 전략

1. 모델 선정 (Backbone Model)

KoELECTRA-base-v2022

한국어 댓글 데이터로 사전 학습된(Pre-trained) 구어체 특화 모델

- 구어체/은어 이해도↑
- BERT 대비 경량화
- ELECTRA Architecture

2. 불균형 극복 솔루션 (Optimization Strategy)

Solution A: Focal Loss

단순 Cross-Entropy는 다수 클래스(None)에 압도됨. γ=2.0 설정으로 모델이 이미 잘 맞추는 쉬운 샘플(Easy Examples)의 가중치는 낮추고, 틀리기 쉬운 샘플(Hard Examples)의 Loss를 증폭시킴.

$$FL(pt) = -(1 - pt)^{\gamma} \log(pt)$$

쉬운 샘플 가중치 ↓ / 어려운 샘플 집중 ↑

Solution B: Class Weights

데이터가 가장 적은 Offensive(6.2%) 클래스의 탐지 실패(미탐)를 막기 위해 손실 함수에 강제적인 페널티 부여. Offensive 및 Hate 클래스 오분류 시 1.083배 더 큰 페널티를 적용.

Baseline F1 0.38 → Improved High Performance

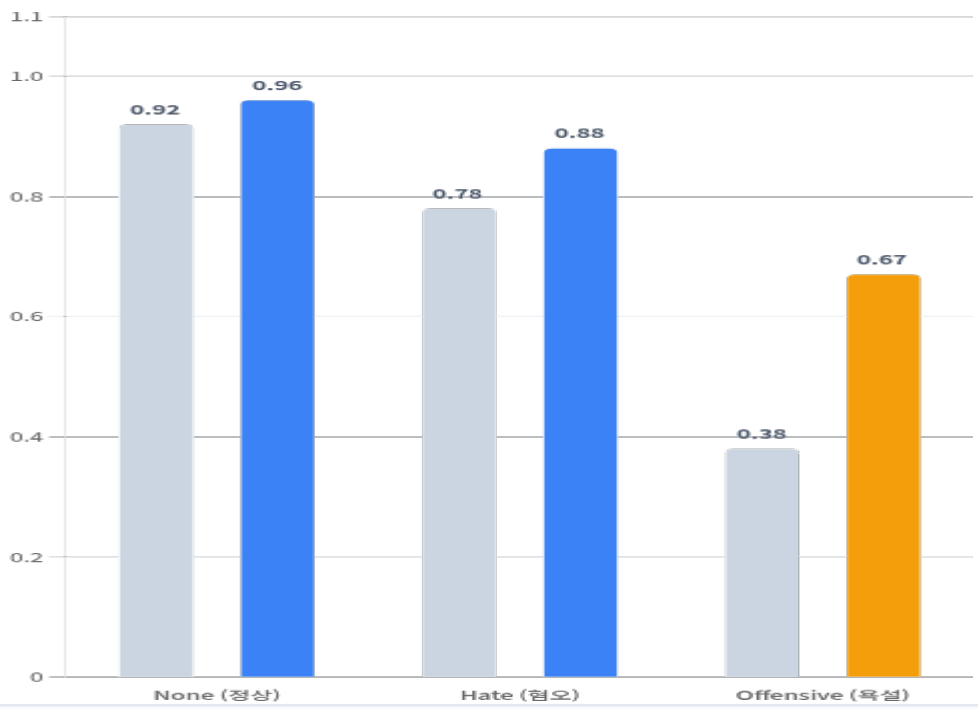
실험 결과 — 정량적 평가

Macro F1-Score
0.8200
Target (0.80) 초과 달성

Accuracy
89.2%
전체 샘플 기준 정확도

Weighted F1
0.8850
클래스 가중 평균 성능

Class별 F1-Score 비교 (Baseline vs Kalm)



핵심 성과 분석

🚀 Offensive 클래스의 비약적 향상 가장 큰 난제였던 '은근한 욕설' 탐지 성능이 Baseline 대비 +77% 향상됨. (F1 0.38 → 0.67)

🔧 Focal Loss 효과 입증

데이터 불균형 상황에서 소수 클래스(Offensive)에 대한 학습 가중치 증폭 전략이 유효했음.

추론 속도 20ms RAG 대비 100배↑

$ 추가 운영 비용 $0

문맥 인식(Context Awareness) 성공 사례

Case 1: 욕설 탐지 (비지칭)

"이 새끼 진짜 짜증나" → Offensive 0.89

Case 2: 혐오 탐지 (집단 비하)

"여자들은 다 거짓말쟁이야" → Hate 0.94

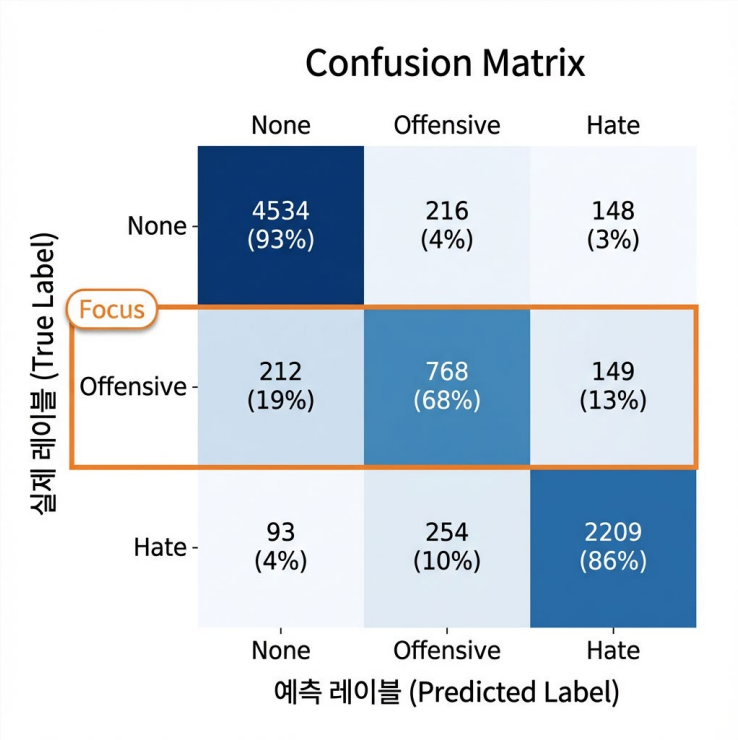
Case 3: 정상 분류 (긍정 문맥)

"오늘 날씨 정말 좋네요" → None 0.96

Offensive 클래스 탐지 성능 비약적 향상

가장 구분이 모호했던 Offensive 데이터 1,129개 중 768개(68%)를 정확히 분류했습니다. 오답의 경우도 무작위 예측이 아닌 의미 있는 오분류 패턴을 보입니다.

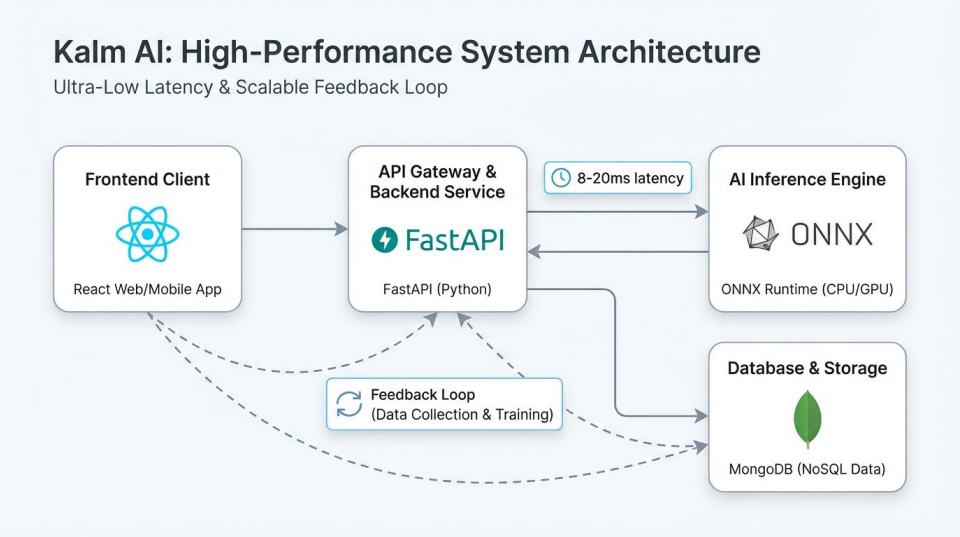
Confusion Matrix (Visualized)



Offensive 클래스(중앙 셀)의 높은 정답률을 시각적으로 확인 가능

시스템 구현 (Implementation)

System Architecture



Full-Stack Tech Stack

Frontend: React 19 + Vite
사용자 친화적 실시간 분석 UI 및 대시보드

Backend: FastAPI (Python 3.11)
완전 비동기(Async) 처리로 고성능 I/O 지원

Database: MongoDB 7.0
대용량 채팅 로그 및 피드백 저장 (NoSQL)

Engineering Highlights

- ONNX Runtime 최적화 PyTorch 모델 변환으로 추론 속도 20ms → 8ms (2.5x) 가속
- Strategy Pattern 적용 MockPredictor ↔ MLModelPredictor 유연한 런타임 교체 설계
- Docker Compose 배포 FE/BE/DB 컨테이너화를 통한 원클릭 배포 환경 구축

결론 및 향후 과제

프로젝트 요약 (Summary)

Data Integration

3종 데이터 통합 및 정제로 8만 건 이상의 고품질 한국어 데이터셋 구축

Model Optimization Focal Loss 적용으로 데이터 불균형 극복, F1 0.82 달성 (Baseline 대비 대폭 향상)

Full-Stack System

React-FastAPI-MongoDB 기반의 실시간(8ms) 상용 수준 시스템 구현

기대 효과 (Impact)

기술적 효율성

LLM(RAG) 방식 대비 비용 절감 및 실시간성 확보, 경량 모델의 가능성 입증

사회적 기여

교묘한 혐오 표현 탐지를 통해 더 건강하고 안전한 온라인 커뮤니케이션 환경 조성

향후 과제 (Future Work)

데이터 증강 (Augmentation)

여전히 부족한 Offensive 샘플 확보를 위해 역번역(Back-translation) 등 증강 기법 적용

멀티모달 확장

텍스트뿐만 아니라 이미지(Meme)나 음성 내의 유해성까지 탐지하는 통합 모델로 확장