HUNTRIX AI

DKTC Threat Detection : TUNiB Hackathon Project

All TEAM MEMBER

TEAM MEMBER

Dongyeun

- Role: Lead AI Researcher & Developer
- Skills:
  - PyTorch, TensorFlow
  - NLP (BERT, GPT)
  - Data Analytics
- Contact:
  idenk9725@gmail.com

Se-Jeong

- Role: Staff AI Visual Designer
- Skills:
  - PyTorch, TensorFlow
  - NLP (BERT, GPT)
  - Data Engineering
- Contact:
  sejong0504@gmail.com

Inha

- Role: Main AI Researcher & Developer
- Skills:
  - PyTorch, TensorFlow
  - NLP (BERT, GPT)
  - Data Engineering
- Contact:
  seoinha4964@gmail.com

Changhun

- Role: Staff AI Researcher & Developer
- Skills:
  - PyTorch, TensorFlow
  - NLP (BERT, GPT)
  - Data Engineering
- Contact:
  stargazer7690@gmail.com

Table of Contents

PROJECT OVERVIEW

TUNiB 해커톤 기업 과제: DKTC 데이터셋 활용
Dataset of Korean Threatening Conversations (한국어 위협 대화 데이터셋)

CORE CHALLENGE:
학습/테스트 간 클래스 불일치(Distribution Shift)로 인해
일반 대화 클래스 과대예측 위험 발생
→ 이를 어떻게 보완할 것인가?

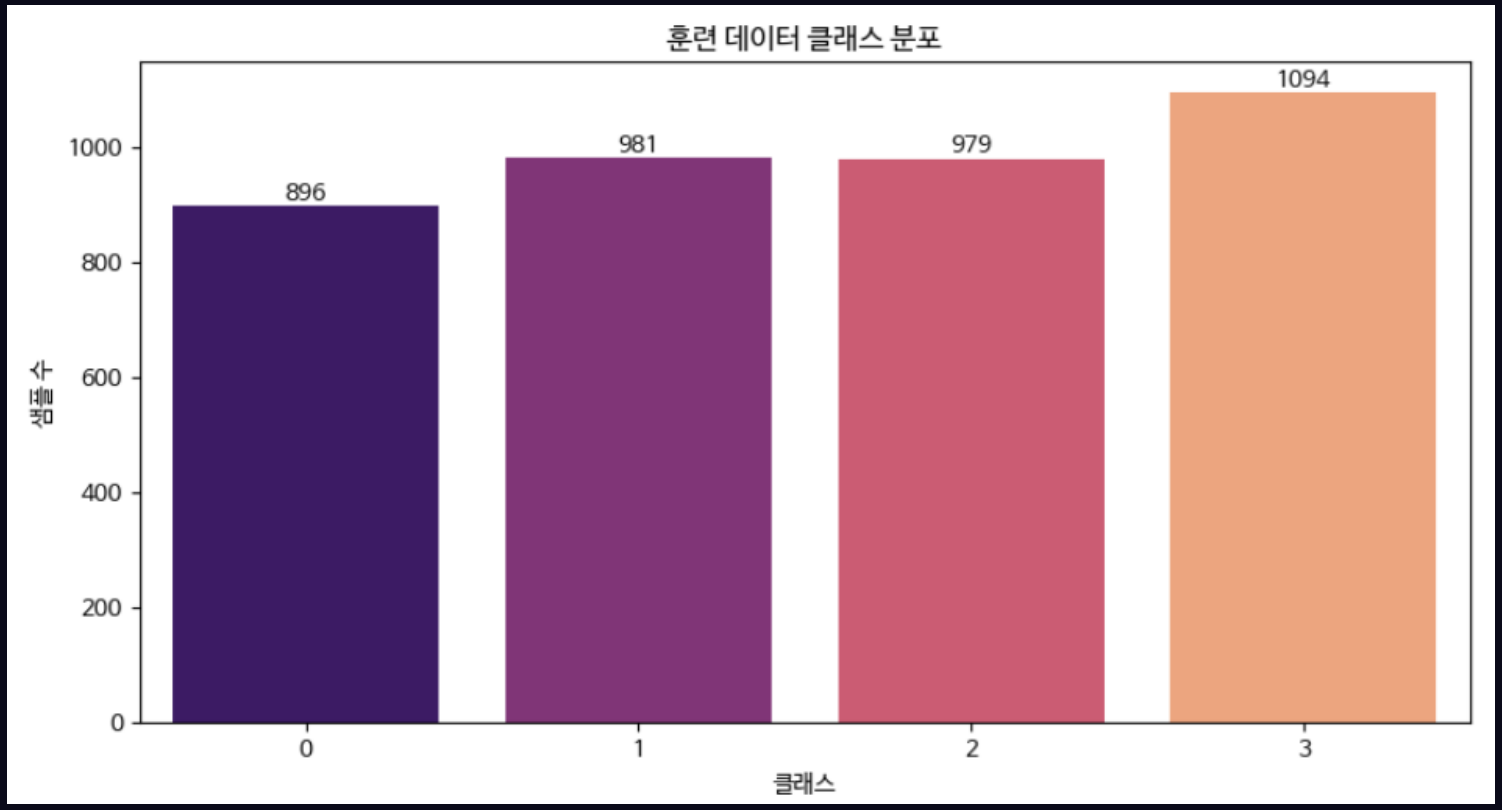
GOAL: 텍스트 다중 분류 모델 최적화 (Max F1-Score)

Data Insight (EDA)

Dataset	협박	갈취	직장 내 괴롭힘	기타 괴롭힘	일반 대화
Train	✓	✓	✓	✓	✗
Test	✓	✓	✓	✓	✓

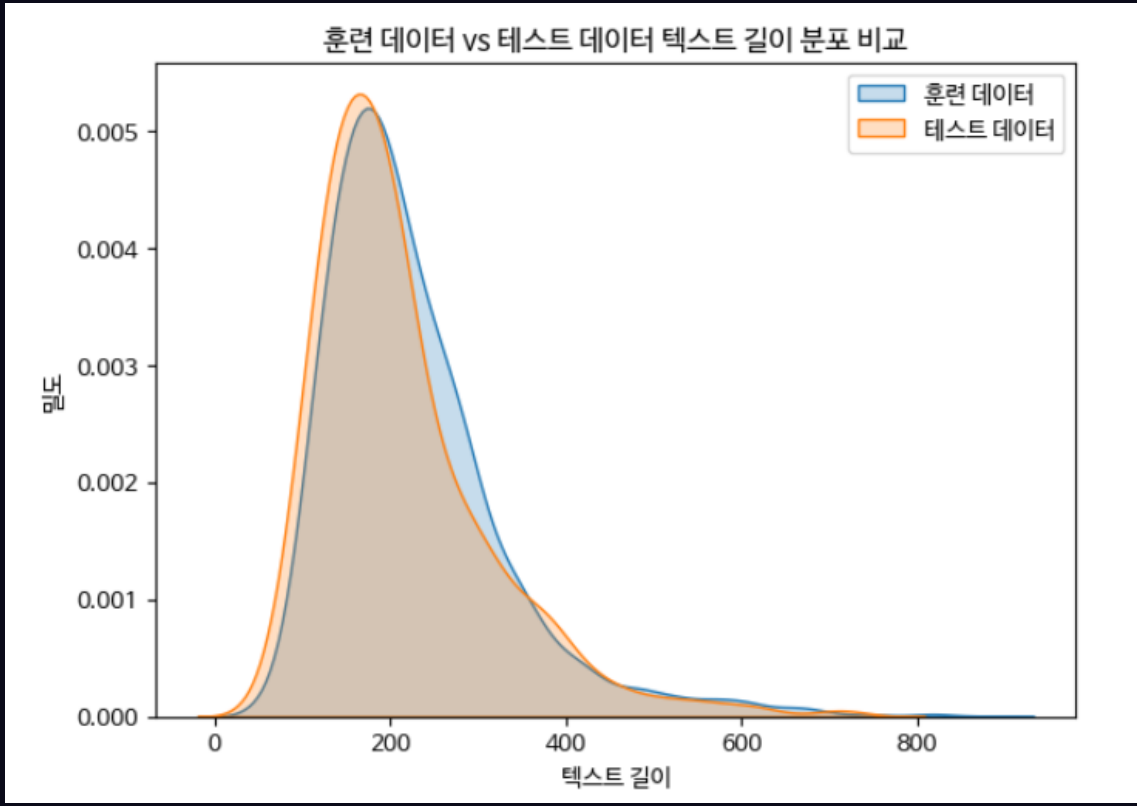
모델은 위협이 아닌 대화 (non-harmful dialogue)를 단 한 번도 학습하지 못한다

Data Insight (EDA)



클래스 간 표본 수 차이는 존재하나, 심각한 불균형 수준은 아니다

Data Insight (EDA)



훈련 데이터와 테스트 데이터의 텍스트 길이 분포는 전반적으로 유사하다

Problem Diagnosis

- **Train 데이터에는 일반 대화 클래스가 존재하지 않음**
- 클래스 간 표본 수는 비교적 균형적
- 길이 분포는 유사

핵심은 일반 대화 클래스의 부재이며, 이를 해결하기 위해 다양한 전략을 실험

Solution

버전	일반대화 조합	모델	epochs	기타 추가 기법	F1-score
v1	없음	beomi/KcELECTRA-base-v2022 (단일 모델)	5	-	0.72
v2	SmileStyle informal(반말) 400개 KakaoChatData(카톡 대화) 300개 kor_unsmile(비혐오 clean=1) 200개 NSMC(긍정 리뷰) 100개 경계 케이스("야 죽을래 ㅋㅋ" 등) 25개 총 1,000개	beomi/KcELECTRA-base-v2022 (단일 모델)	5	K-Fold CV (5-Fold)	0.74
★ **v3**	SmileStyle(개별 문장) 1,200개 kor_unsmile(개별 문장) 800개 KakaoChatData (파싱 오류로 건너뜀) NSMC(개별 문장) 500개 Hard Negative 경계 대화 165개 (HN-A/B/C/D 각 50개) **총 2,665개**	beomi/KcELECTRA-base-v2022 klue/bert-base	5	기본 인프라 K-Fold CV (5-Fold) LLRD EMA FGM (Adversarial Training) 학습 기법 (Ablation Study) Focal Loss R-Drop 추론 단계 후 처리 Prior Shift Calibration Hard Negative Mining (SBERT) Dynamic Class Weight Confidence Fallback Cosine Dedup + Quality Filter	**0.79**
v4	SmileStyle(개별 문장) 1,200개 kor_unsmile(개별 문장) 800개 KakaoChatData(Q+A 쌍) 500개 NSMC(개별 문장) 500개 korean_safe_conversation 10,000개 kor_nli 한국어 문장 10,000개 Hard Negative 경계 대화 165개 **총 23,165개**	일반대화 데이터로 사전학습시킨 klue/bert-base (TAPT 적용)	5	기본 인프라 K-Fold CV (5-Fold) 학습 기법 (Ablation Study) Focal Loss R-Drop 추론 단계 후 처리 (학습과 분리됨) LLRD EMA FGM 추가적 사전학습 TAPT 적용	0.72

Leaderboard Score flow

주요 결정 및 결과

- 결정1: Korean safe conversation 1,000개를 수집하고, K-Fold를 적용
- **결과**: F1=0.74 (+0.02) 개선됐지만, 문장을 2~3개 concatenation 방식이 부자연 스러움

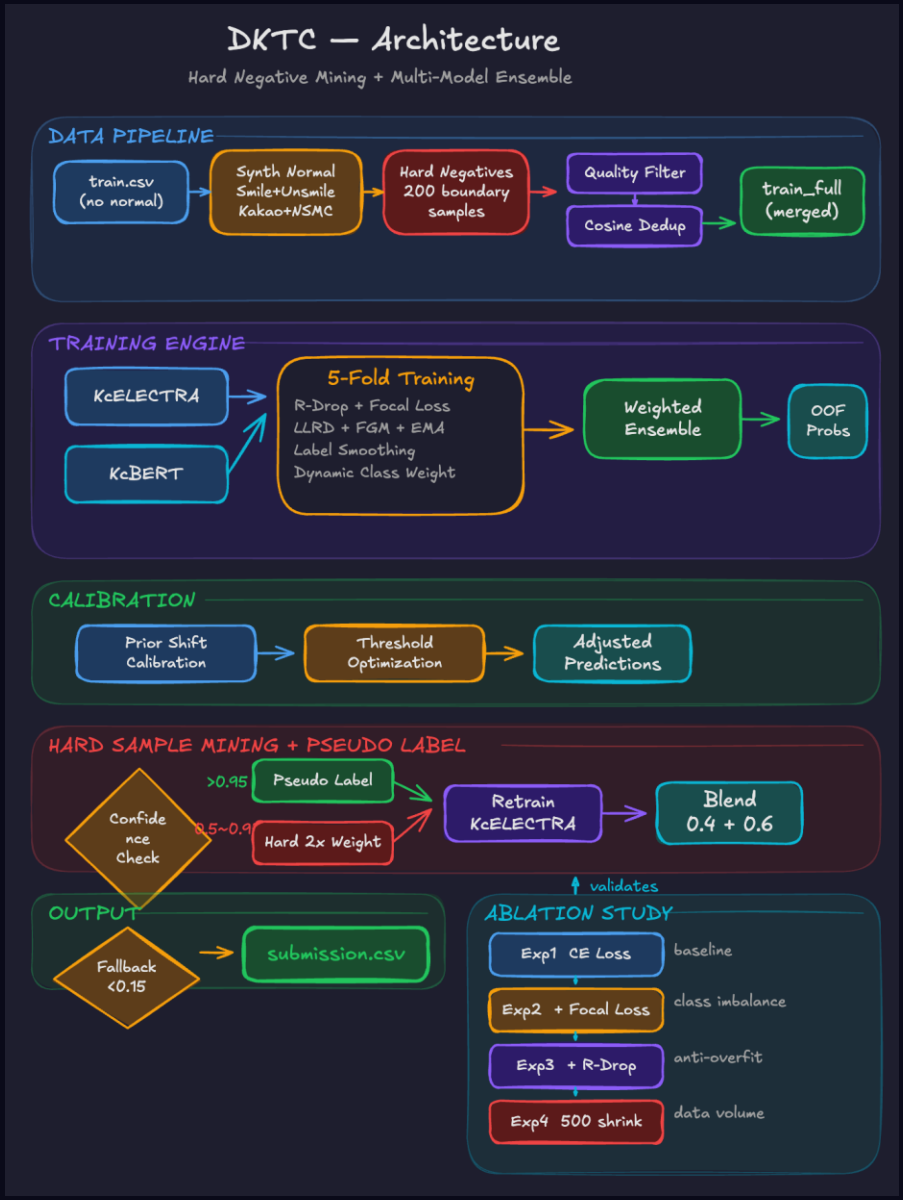
- 결정1: 개별 문장 샘플링으로 전환
- 결정2: SBERT로 전체 텍스트 임베딩 후, 위험 대화와 cosine similarity 높은 일반 대화 200개를 선별 (Hard Negative Mining)
- 결정3: 수업에서 배운 Focal Loss, LLRD, EMA와 논문 기반 FGM, R-Drop, Prior Shift Calibration을 종합 적용

- **결과**: F1=0.79 (+0.05). **Hard Negative Mining이 가장 큰 기여**

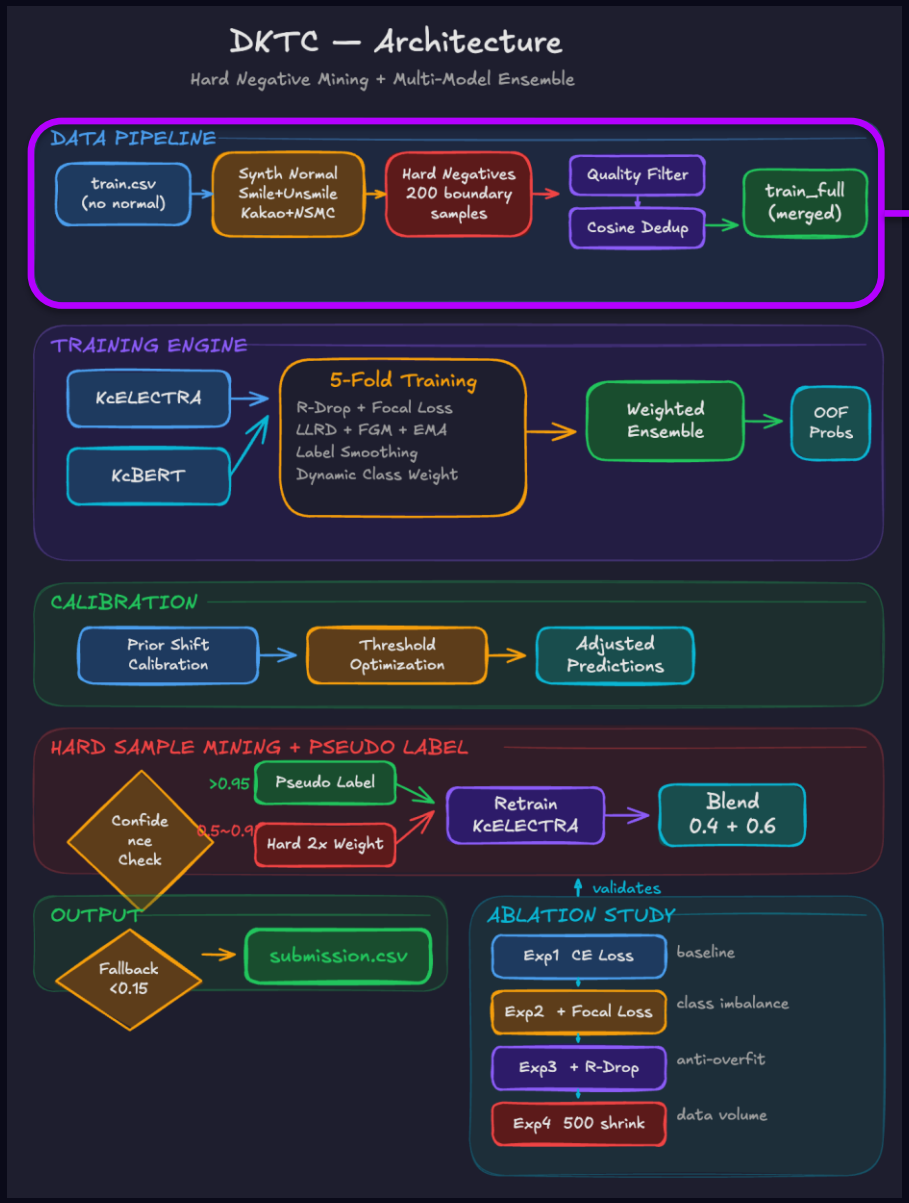
- 결정1: TAPT를 위해 BERT 구조인 klue/bert-base로 모델 변경
- 결정 2: 데이터를 대폭 늘려 더 다양한 패턴을 학습시켜 Korean safe conversation 10,000개 + kor_nli 10,000개

- 결과 : F1=0.72 **오히려 v1 수준으로 하락**

Preprocessing & Model Architecture



Preprocessing & Model Architecture



1단계: DATA PIPELINE — 데이터 수집 + 정제

- 원본 train.csv에는 "일반 대화" 클래스가 0개라서 외부에서 합성해야 함
- SmileStyle, kor_unsmile, KakaoChat, NSMC 4개 소스에서 일반 대화 약 3,000개 수집
- Hard Negative 200개 추가 — "죽여버린다 ㅋㅋ 이 게임 보스가~" 같이 괴롭힘 처럼 보이지만 실제로는 일반 대화인 경계 샘플
- Quality Filter → 너무 짧거나, 특수문자 과다, 반복 패턴 제거
- Cosine Dedup → 임베딩 유사도 0.95 이상인 중복 샘플 제거
- 결과: 정제된 train_full 완성

Configuration: Data Synthesization

일반 대화 데이터 생성 및 합성

- SmileStyle(개별 문장) 1,200개
- kor_unsmile(개별 문장) 800개
- NSMC(개별 문장) 500개
- 경계선 대화165개

총 2,665개

선정 이유

- 오탈자와 스타일 변환 오류를 내포하여 노이즈 추가 가능
- 욕설, 모욕 등 경계선 데이터 다수
- 소스 다양화

Preprocessing & Model Architecture

2단계: TRAINING ENGINE — 2모델 × 5Fold 학습

- KcELECTRA + KcBERT 두 모델을 각각 5-Fold 교차검증
- 학습 기법 총 7개 적용:
  - Focal Loss — 소수 클래스에 더 집중
  - R-Drop — 같은 입력 2번 통과시켜 과적합 방지
  - LLRD — 아래 층은 천천히, 위 층은 빠르게 학습
  - FGM — 임베딩에 노이즈 줘서 적대적 훈련
  - EMA — 파라미터 이동평균으로 안정적 학습
  - Label Smoothing — 정답에 약간의 불확실성 부여
  - Dynamic Class Weight — 테스트 분포 추정치 기반 가중치
- 10개 Fold 결과를 Val F1 성능 기반 가중 앙상블

Configuration: Model

선정 이유

- beomi/KcELECTRA-base-v2022

  온라인 댓글로 사전학습된 Transformer 계열 모델로, 문장의 흐름을 잘 파악함

- beomi/kcbert-base

  온라인 댓글로 사전학습된 BERT 계열 모델로, 단어의 의미를 잘 파악함

Configuration: Model Tuning

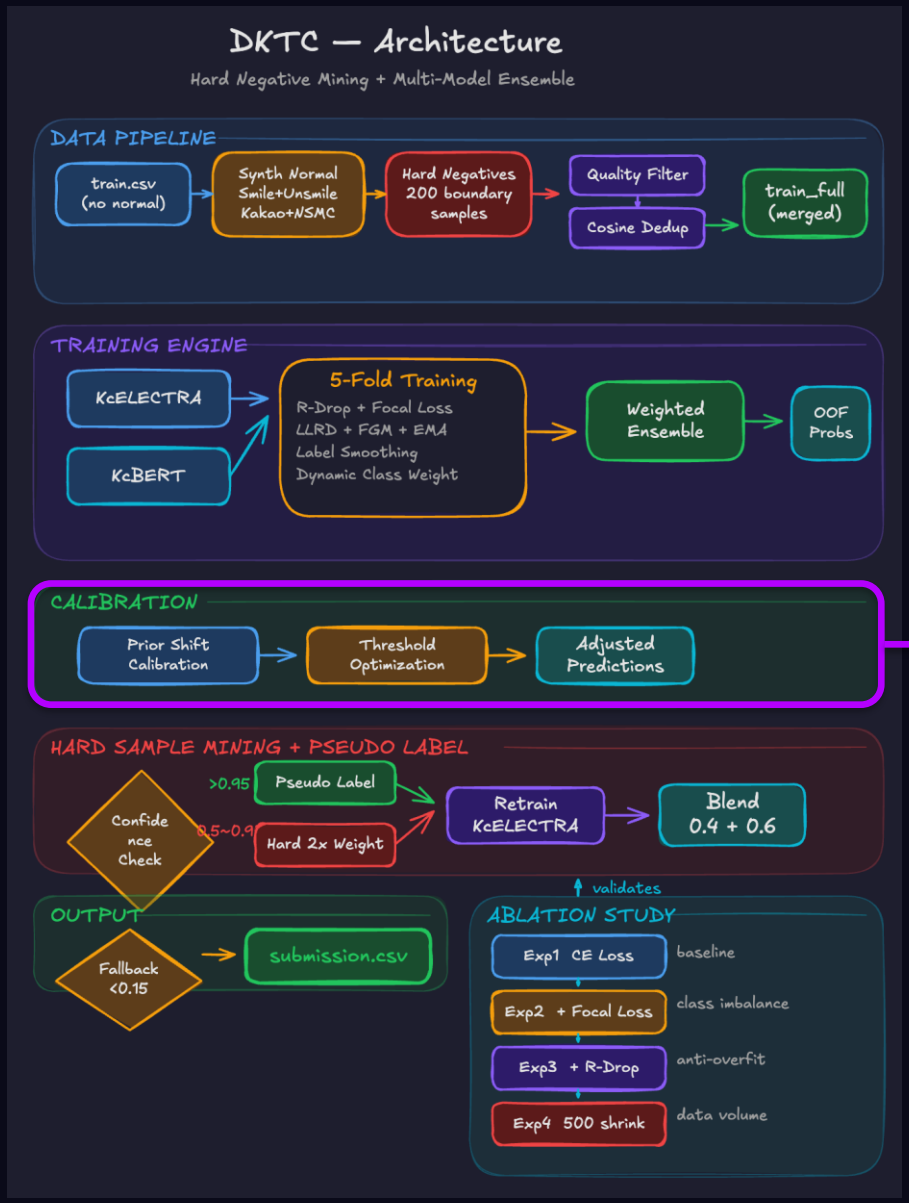
선정 이유

- R-Drop
  동일한 데이터에 대한 두 번의 드롭아웃 결과물 사이의 일관성을 강제하여, 모델의 출력 분포를 정규화하고 강건성(Robustness)을 높이기 위해 선택

- Focal Loss
  정답을 맞히기 쉬운 데이터의 비중을 낮추고 학습하기 어려운(Hard) 데이터에 가중치를 부여함으로써, 데이터 불균형 문제를 해결하기 위해 선택

- Hard Negative Mining
  모델이 오답으로 분류하기 쉬운 까다로운 오답 사례(Hard Negative)를 집중적으로 학습시켜, 클래스 간 경계선을 정교하게 다듬기 위해 선택

Preprocessing & Model Architecture



3단계: CALIBRATION — 예측 확률 보정

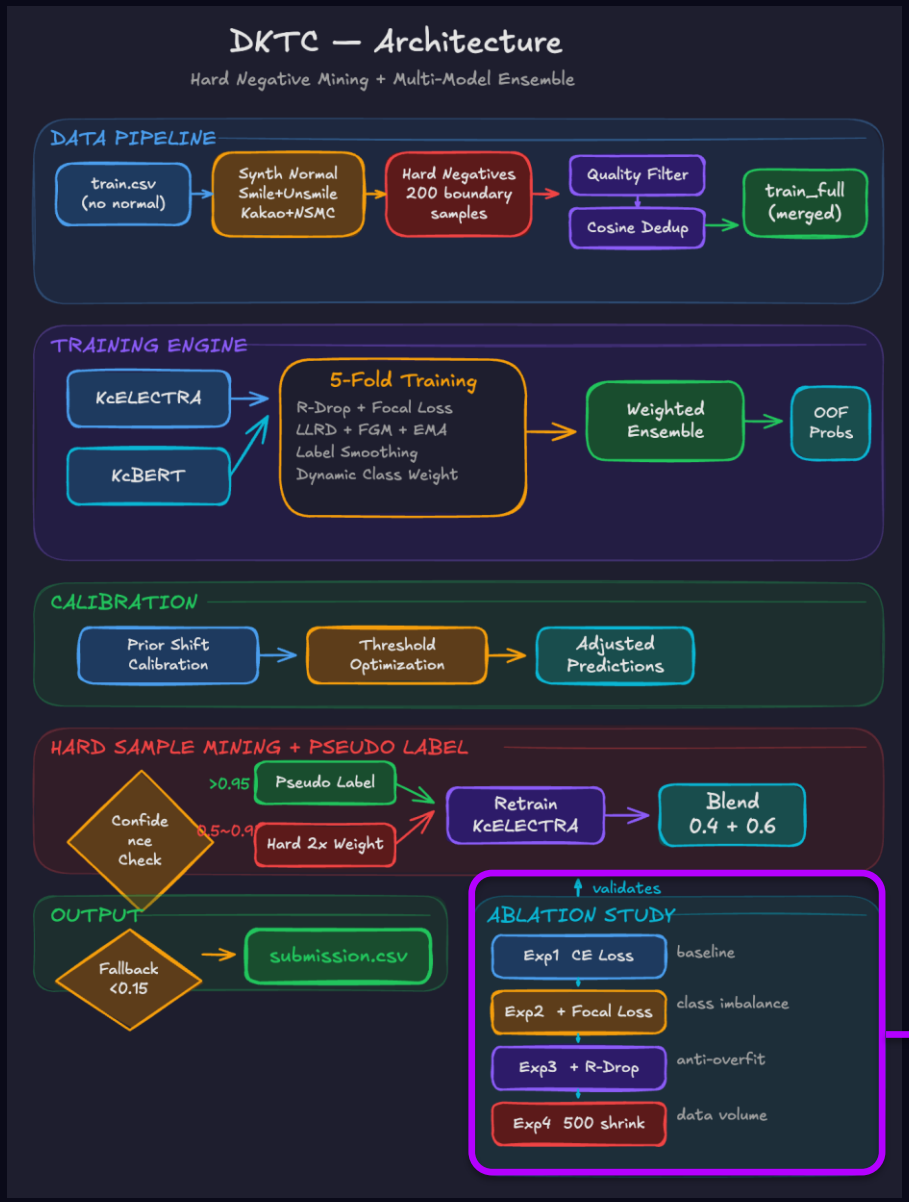
- Prior Shift Calibration — train 분포 ≠ test 분포이므로, 추정 test 분포 비율로 확률 재조정
- Threshold Optimization — 클래스별 최적 임계값을 OOF (Out of Fold) 데이터에서 탐색
- 결과: 분포가 보정된 Adjusted Predictions

Preprocessing & Model Architecture

4단계: HARD SAMPLE MINING — 어려운 샘플 재학습

- 확신도 ≥ 0.95 → Pseudo Label로 학습 데이터에 추가
- 확신도 0.5~0.95 → Hard Sample로 2배 복제해서 가중 학습
- 확신도 < 0.5 → 불확실하므로 제외
- 확신도 < 0.15인 샘플은 Fallback → "일반 대화" 로 강제 배정
- 이 데이터로 KcELECTRA를 한 번 더 재학습
- 기존 앙상블과 재학습 모델을 0.4 : 0.6 비율로 블렌딩

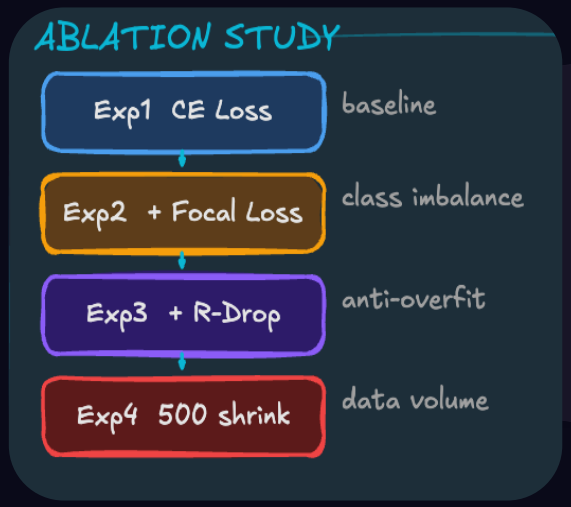
Preprocessing & Model Architecture



Ablation Study — 각 기법의 기여도 검증

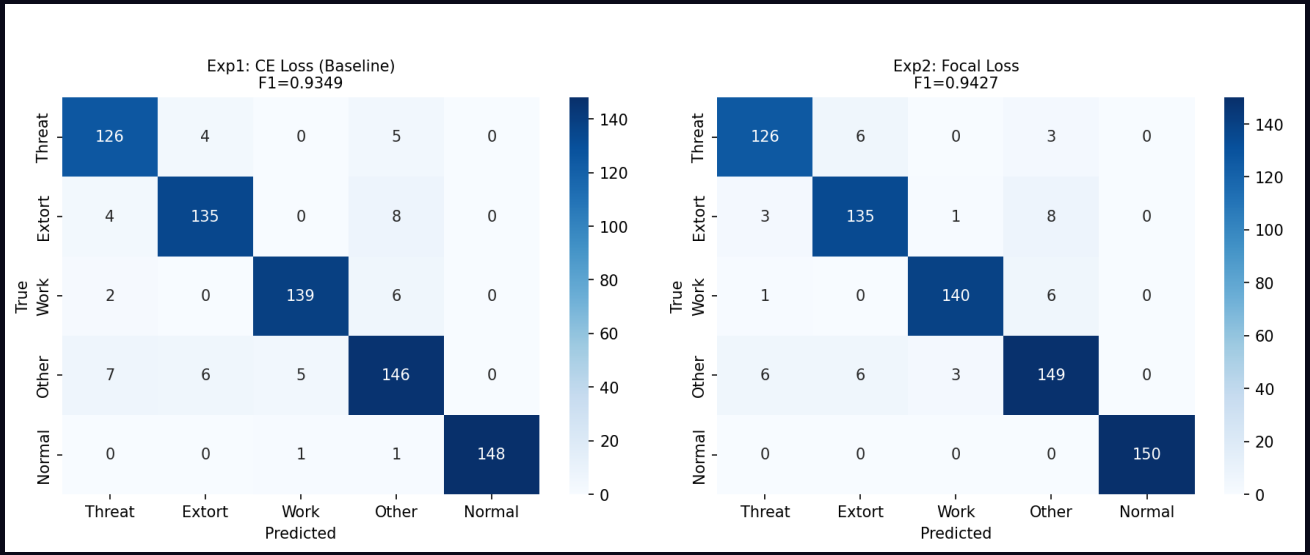
- Exp1 CE Loss만 → baseline 성능 측정
- Exp2 + Focal Loss → 클래스 불균형 해결 효과 확인
- Exp3 + R-Drop → 과적합 방지 효과 확인
- Exp4 합성 데이터 500개로 축소 → 데이터 양이 성능에 미치는 영향 확인

Ablation Study - v3

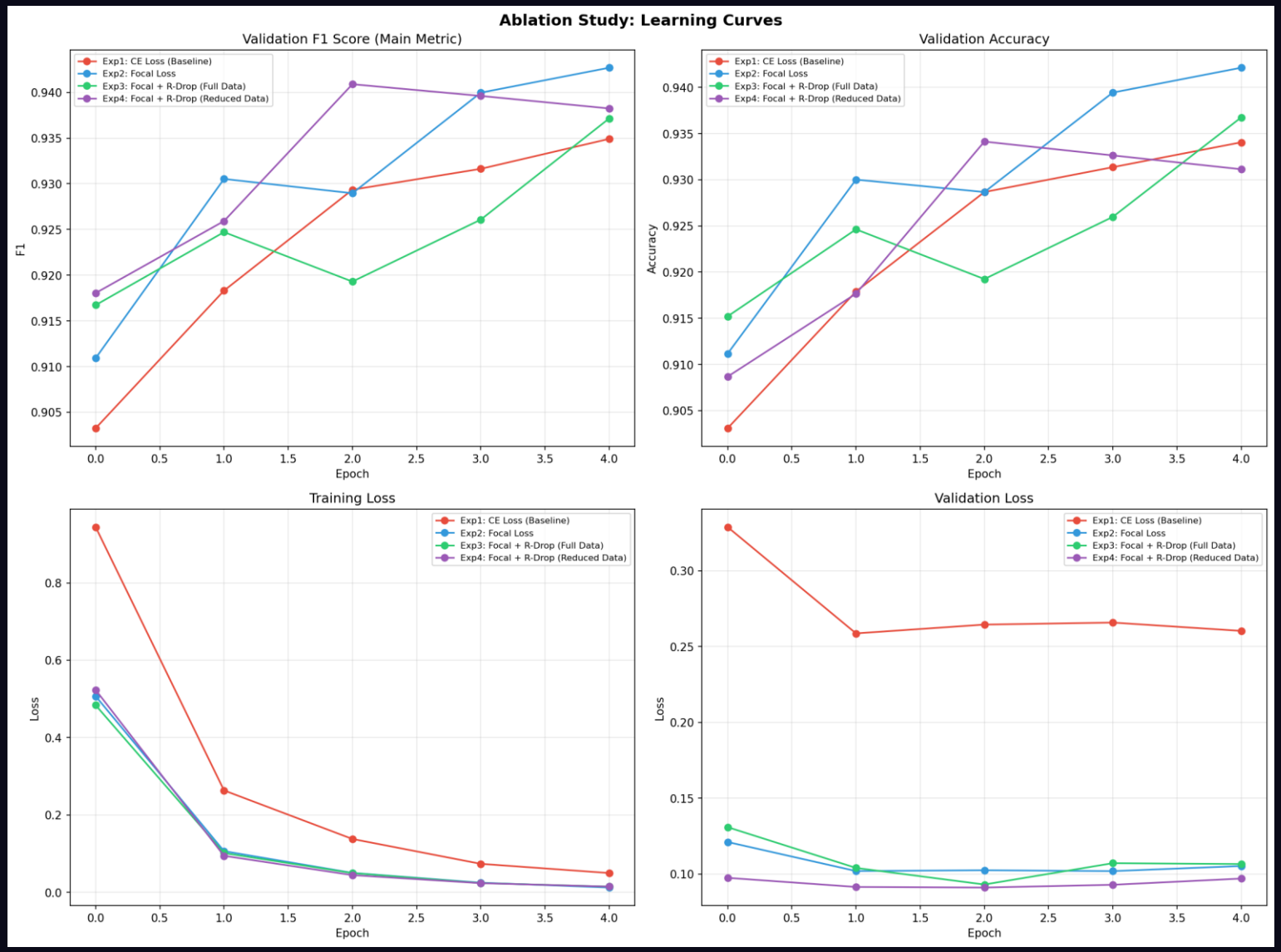


	실험 사유	예상 효과
+ Focal Loss	클래스 불균형 상황에서 CE는 쉬운 샘플에도 동일 가중치. Focal Loss는 잘 맞추는 샘플 가중치↓, 어려운 샘플↑ → 소수 클래스 학습 효율 향상	일반 대화(다수)는 빠르게 학습 후 가중치 감소, 위험 대화(소수) 분류에 학습 집중
+ R-Drop	같은 입력 두 번 forward pass → 서로 다른 dropout mask로 다른 출력 발생. KL divergence로 일관된 예측 강제 (self-distillation)	다른 regularization과 함께 과적합 방지 기여
+ 합성데이터 500개 축소	합성 데이터 양이 모델 성능에 영향	소량 데이터양에 따라 성능 개선 기여

Ablation Study - Result



Ablation Study - Result



Conclusion

- 일반 대화 데이터 추가는 모델의 결정 경계를 안정화하는 데 기여
- 성능 향상의 가장 큰 기여 요소는 단연 Hard Negative Mining
- 단순 TAPT 적용은 오히려 성능 저하를 유발
- 과도한 사전학습보다 데이터 품질과 경계 샘플 설계가 더 중요

핵심은 모델 구조 변경이 아니라, 데이터 구조 보완과 경계 샘플 설계에 있었다

프로젝트 제출 평가 기준 답변

1. 데이터 EDA와 데이터 전처리가 적절하게 이뤄졌는가?

- 합성 일반대화 생성 (6개 소스 ~3k)
- Hard Negative Mining (경계 케이스 200개)
- 품질 필터 (길이/특수문자)
- 코사인 유사도 기반
- 중복 제거

2. task에 알맞게 적절한 모델을 찾아보고 선정했는가?

- Korean-specialized models (KcELECTRA, KcBERT) — 한국어 사전학습

3. 성능향상을 위해 논리적으로 접근했는가?

- 기본 인프라
  - K-Fold (5-fold)
  - LLRD (layer-wise LR decay)
  - FGM (adversarial training)
  - EMA
- 학습 기법
  - Focal Loss
  - R-Drop
- 예측 확률 보정 및 샘플 재학습
  - Label Smoothing
  - Weighted Ensemble
  - Threshold Optimization

프로젝트 제출 평가 기준 답변

4. 결과 도출을 위해 여러 가지 시도를 진행했는가?

- v1→v2→v3→v4: 점진적 개선
- v3에서 Hard Negative Mining
  - 확신도 ≥ 0.95 → **Pseudo Label**로 학습 데이터에 추가
  - 확신도 0.5~0.95 → **Hard Sample**로 2배 복제해서 가중 학습
  - 확신도 < 0.5 → 불확실하므로 제외
  - 이 데이터로 KcELECTRA를 **한 번 더 재학습**
  - 기존 앙상블과 재학습 모델을 **0.4 : 0.6 비율로 블렌딩**
- 각 버전마다 체크포인트 저장으로 런타임 안정성 확보

5. 도출된 결론에 충분한 설득력이 있는가?

- - 평균 Val F1 ~0.94, v8: 평균 Val F1 ~0.9425
- - 논문 근거: Focal Loss (Lin 2017), R-Drop (Liang 2021), FGM (Miyato 2017)

6. 적절한 metric을 설정하고 그 사용 근거 및 결과를 분석하였는가?

- Macro F1 (클래스 불균형 고려), Accuracy (보조 지표)
- OOF (Out-of-Fold) 검증으로 과적합 방지
- 클래스별 threshold 최적화로 precision/recall 균형

Retro

동연

- 우선 굳이 뭘 하라고 하지 않아도 알아서 일을 나누고 진행한 팀원 분들께 진심으로 감사드린다.
- 팀의 키워드는 "자유", 모토는 "하고 싶은 거 다 하자"로 삼았는데, 지난 3일 동안 정말로 스스로 질문하고 궁금한 부분을 실험으로 구현할 수 있었던, 모토에 잘 맞는 시간을 보낸 것 같아 후회는 없다.
- 가장 아쉬웠던 지점을 꼽자면 모델 학습에 오랜 시간이 걸리는 점이었다. 해보고 싶은 게 아직 많은데, 모델 학습에 시간이 오래 걸려 시도할 수 있었던 성능 개선 기법의 수가 비교적 적다. 체크포인트 등 학습 속도를 빠르게 병렬적으로 할 수 있는 방법으로 어떤 것들이 있는지 더 알아봐야겠다.

인하

- 데이터 스케일링보다 경계 사례 선별이 모델 성능을 좌우한다.
- 데이터는 양이 아니라 질이다. v4에서 데이터를 10배로 늘렸는데 오히려 성능이 떨어졌다. 도메인이 안 맞는 대량 데이터보다 도메인에 맞는 소량의 고품질 데이터가 훨씬 효과적이었다.
- 가장 큰 성능 개선은 Hard Negative Mining에서 나왔다. 일반 대화를 많이 넣는 것보다, 위험 대화와 혼동되기 쉬운 경계 사례를 골라서 학습시키는 게 결정적이었다.

세종

- 이번 프로젝트에서 가장 큰 수확은 단연 팀워크였다. 아직 딥러닝에 익숙하지 않아 기술적인 벽에 부딪힐 때마다 친절하게 설명해주고, 기다려준 팀원들 덕분에 끝까지 포기하지 않고 시도해볼 수 있었다. 성능 지표를 올리는 것을 넘어 팀원들 간 토론하고, 고민을 나누는 과정이 재미를 느끼게 해준 소중한 경험이었다.

창훈

- 각자 강점에 맡게 역할 분담해주신 팀원분들께 감사합니다.
- End-to-End를 경험 할 수 있게 되어 전체적인 데이터로 문제 해결에 대한 방향을 잡을 수 있는 값진 기회였습니다.


HuntrixAI
Thank You