

DKTC

(Dataset of Korean Threatening Conversations)

다중 분류

2025년 5월 7일

리서치 온라인 13기 <허씨 양반김>

멤버 : 김영숙, 반태훈, 양지웅

프로젝트 개요

0. 프로젝트 목적

I. 모델 성능 비교

- From-Scratch Model
 - 밑바닥부터 학습한 모델
 - BiLSTM
- Pre-trained model(fine-tuning)
 - 사전학습된 모델을 파인튜닝
 - KcELECTRA
 - KcBert

II. 데이터 증강에 따른 성능 비교

- KoEDA

III. 일반대화 데이터셋에 따른 성능 비교

- LLM 프롬프트로 생성한 데이터셋
- AI hub의 일반 데이터셋

1. Pre-trained Model 조사

모델명	구조	학습 방식	특징	분류 작업 적합도
KcBERT	BERT 기반	Masked LM	한국 커뮤니티(욕설, 비속어 포함) 데이터로 학습됨 → 욕설 탐지, 감정 분석에 강함	★★★★★
KoDialoGPT	GPT 기반 (Decoder-only)	Auto-regressive LM	대화 생성 특화 모델 (응답 생성, 챗봇 용도)	★☆☆☆☆
KoGPT	GPT 기반 (Decoder-only)	Auto-regressive LM	문장 생성에 강하나 분류에는 구조적으로 부적합	★☆☆☆☆
KoBERT	BERT 기반	Masked LM + NSP	안정적인 분류 성능, 한국어 전용 토큰나이저 내장	★★★★★
KcELECTRA	BERT 구조 + ELECTRA 방식	Replaced Token Detection	빠르고 정확함, 분류에 가장 최적화된 구조	★★★★★

1. Pre-trained Model 조사

- ❑ 문장 생성, AI 챗봇 → KoGPT
- ❑ 감정 분석, 문장 분류 → KoBERT
- ❑ 빠르고 정확한 분류 모델 필요 → KoELECTRA
- ❑ KcELECTRA : 속도와 정확도 모두 우수 → 1순위
- ❑ KoBERT : 안정적이고 벤치마크 많음 → 무난한 성능
- ❑ KcBERT : 한국 커뮤니티 언어 특화 → 비속어/공격성 탐지에 유리
 - Layer 구성 (BERT-base 기준)
 - 12개의 Transformer Encoder Layer
 - 각 Layer마다 Multi-head Attention + FFN
 - Hidden size = 768
 - 총 파라미터 수 약 110M

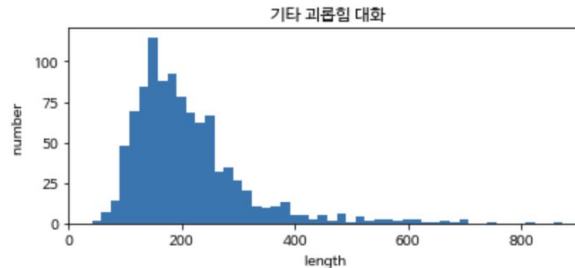
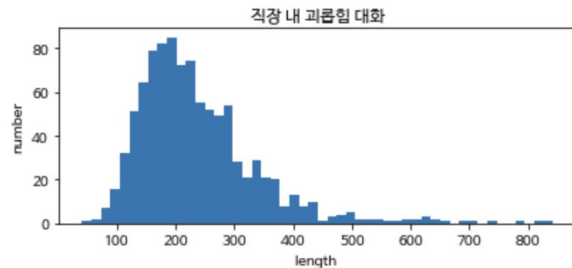
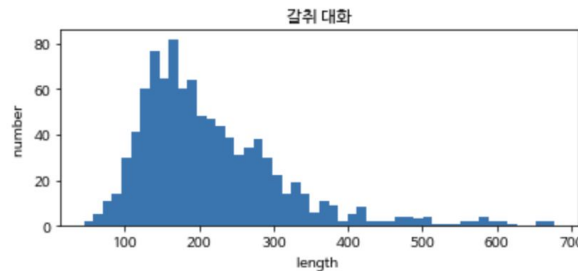
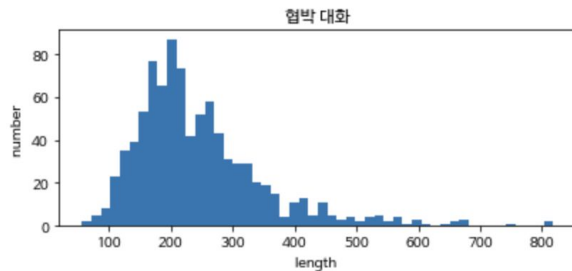
프로젝트 진행

2. EDA(Exploratory Data Analysis)

Tunib DKTC Dataset

idx	class	conversation
0	0	협박 대화 지금 너 스스로를 죽여달라고 애원하는 것인가?\n 아닙니다. 죄송합니다.\n 죽을 ...
1	1	협박 대화 길동경찰서입니다.\n9시 40분 마트에 폭발물을 설치할거다.\n네?\n꼭바로 들어 ...
2	2	기타 괴롭힘 대화 너 되게 귀여운거 알지? 노바도 작은 남자는 침봤어.\n그만해. 니들 놀리는거 재미...
3	3	갈취 대화 어이 거기\n에??\n너 말이야 너. 이리 오라고\n무슨 일.\n너 웃 좋아보인다?...
4	4	갈취 대화 저기요 혹시 날이 너무 뜨겁잖아요? 저희 회사에서 이 선크림 파는데 한 번 손등에 ...

클래스	Class No.	# Training	# Test
협박	00	896	100
갈취	01	981	100
직장 내 괴롭힘	02	979	100
기타 괴롭힘	03	1,094	100
일반	04	-	100



항목	값
전체 샘플 수	3950개
평균 길이	약 226.57자
표준편차	약 104.56자
최소 길이	41자
1사분위수 (25%)	156자
중앙값 (50%)	203자
3사분위수 (75%)	270자
최대 길이	874자

2. EDA(Exploratory Data Analysis)

중복 및 결측치 확인

- 중복 데이터

- 클래스와 **Conversation** 모두 중복되는 데이터 104건

```
orig_train의 전체 샘플수 : 3950
orig_train의 conversation 열에서 중복을 배제한 유일한 샘플의 수 : 3846
orig_train의 중복제거 샘플수 : 104
```

- 중복 데이터 예시

```
idx class conversation
3025 3025 직장 내 괴롭힘 대화 저 자식 식판에 밥 푸는 거 봐라. 무식한 거 봐.\n옛날에 태어났으면 딱 머슴할 ...
3304 3304 직장 내 괴롭힘 대화 저 자식 식판에 밥 푸는 거 봐라. 무식한 거 봐.\n옛날에 태어났으면 딱 머슴할 ...
```

- 중복 데이터 **Class**별 개수

기타 괴롭힘 대화 166 직장 내 괴롭힘 대화 18 갈취 대화 16 협박 대화 8

- 중복 데이터 제거 후 **Class**별 개수
총 3846건

클래스	중복 제거 전	중복 제거 후
협박	896	892
갈취	981	973
직장 내 괴롭힘	979	970
기타 괴롭힘	1,094	1,011

- 결측 데이터

- 클래스와 **Conversation**에 결측 데이터는 없음

2. EDA(Exploratory Data Analysis)

클래스별 빈출 단어

협박 대화	갈취 대화	직장 내 괴롭힘 대화	기타 괴롭힘 대화
제발, 해, 진짜, 지금, 사람	돈, 좀, 진짜, 그럼	죄송합니다, 대리, 회사, 일	각 클래스의 빈출 단어와 유사

갈취

- "돈"에 대한 TF-IDF 점수가 매우 높게 나타남
- **금전 요구**라는 핵심 특징을 모델이 잘 학습할 경우, **높은 분류 성능** 기대됨

직장 내 괴롭힘

- **직급 언급** (예: 대리, 과장) 및 **직장 내 표현**이 빈번하게 등장
- 맥락이 명확히 드러나므로, **모델이 분류하기 용이** 할 것으로 예상됨

협박 & 기타 괴롭힘 대화

- 두 클래스 모두 **공통적으로 사용되는 단어**가 많음
- 모델이 구분하기 어려워 **분류 성능 저하** 가능성 있음
 - 두 클래스 간 특징 차이 분석 필요

2. EDA(Exploratory Data Analysis)

협박 & 기타 괴롭힘 클래스 간 차별적 단어 추출

1. 각 클래스 별로 등장한 단어들을 집계
2. 자주 등장한 n개 단어 중에서, 상대 클래스에 비해 등장 비율이 2~3배 이상 높은 단어들만 필터링

협박 클래스에서 두드러지는 단어

- 칼: 협박(165회), 기타 괴롭힘(2회) -> 82.5%
- 죽여: 협박(249회), 기타 괴롭힘(4회) -> 62.2%
- 살려주세요: 협박(142회), 기타 괴롭힘(6회) -> 23.6%

→ 협박 대화에서는 목숨과 관련된 폭력적인 단어가 자주 등장

기타 괴롭힘 클래스에서 두드러지는 단어

- 장애인: 기타 괴롭힘(98회), 협박(2회) -> 49.0%
- 고객: 기타 괴롭힘(173회), 협박(5회) -> 34.6%
- 돼지: 기타 괴롭힘(111회), 협박(4회) -> 27.7%
- 냄새: 기타 괴롭힘(139회), 협박(7회) -> 19.8%

→ 기타 괴롭힘 대화에서는 비하/차별 관련 단어가 자주 등장

3. 일반 대화 데이터 셋 구축 방법 - LLM 모델

LLM 모델(Chat GPT, Claude) 사용

프롬프트 예시

[입력 데이터]:

"0": "지금 너 스스로를 죽여달라고 애원하는 것인가?\n아닙니다. 죄송합니다.\n죽을 거면 죽
"1": "길동경찰서입니다.\n9시 40분 마트에 폭발물을 설치할거다.\n네?\n꼭바로 들어 한번만
"2": "너 되게 귀여운거 알지? 나보다 작은 남자는 쳤잖어.\n그만해. 니들 놀리는거 재미없어.
"3": "어이 거기\n예??\n너 말이야 너. 이리 오라고\n무슨 일.\n너 웃 좋아보인다?\n애 돈

[프롬프트]:

"위 문장들은 협박,갈취,직장 내 괴롭힘,기타 괴롭힘 중 하나에 해당하는 대화입니다.
위 문장들과 유사한 환경에서 나올만한 일상적인 대화로 만들어주세요."

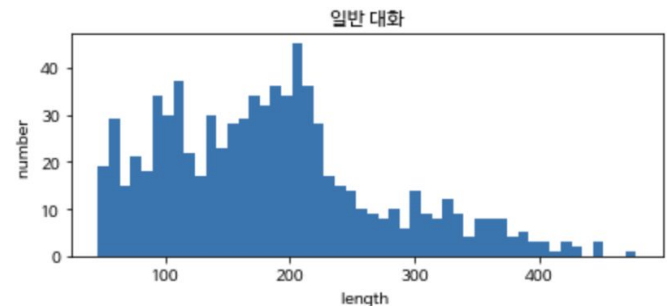
요구사항:

- 문장 길이는 평균 226, 표준편차 104의 정규분포로 생성
- class는 "일반 대화"
- 생성 문장 수는 50개
- 발화자 사이에는 \n으로 구분

[출력 예시]:

```
[
  {
    "class": "일반 대화",
    "conversation": "요즘 날씨가 참 좋네"
  },
  {
    "class": "일반 대화",
    "conversation": "오늘 저녁 뭐 먹을래?"
  }
]
```

정책에 위반되는 내용은 포함하지 말아주세요.



생성 데이터 예시

```
[
  {
    "class": "일반 대화",
    "conversation": "오늘 점심 뭐 먹었어?\n회사 근처 새로 생긴 국수집에서 냉면 먹었"
  },
  {
    "class": "일반 대화",
    "conversation": "이번 주말에 뭐해?\n특별한 계획은 없어. 집에서 쉴 것 같아.\n글"
  },
  {
    "class": "일반 대화",
    "conversation": "커피 한잔할래?\n좋아, 근처에 괜찮은 카페 알아?\n여기서 2분 가"
  },
]
```

3. 일반 대화 데이터 셋 구축 방법 - AI HUB

AI HUB 오픈 데이터 사용

- [주제별 텍스트 일상 대화 데이터](#) 사용
- 카카오톡, 페이스북, 인스타그램, 밴드, 네이버온에서 확보한 일상 대화 데이터
- 원하는 형태로 파싱하기 쉬운 형태

데이터 예시

1 : 회사 후임이 너무 안 뽑힌다.
2 : 네가 성격이 너무 썩어서 그런 거 아니냐? ㅋ
1 : 아니야. 성질 전혀 안 부렸어. 내가 얼마나 잘 해 주는데 ㅋ
2 : 아예 사람이 안 뽑히는 거야?
1 : 아니 뽑히긴 하는데, 지금 4번째인가 갈아 치웠나 보다. ㅠ
2 : 응? 그런 식으로 많이 관렸다고?
2 : 그럼 99프로 선임 문제 아니냐? ㅋ
1 : 놈! 일주일 하다가 관둔 애들도 많고, 한 달 정도 하다가 업무 실수하고 업무 너무 어렵다고 관둔 사람도 있다.
2 : 요샌 왜 이렇게 참을성들이 없는지...
1 : 그 일 내가 다 독박 써서 하고 있다.
2 : 2명이 할 일을 혼자 하다니? 엄청 힘들겠네.
1 : 응 그래서 매일 야근에, 주말에 특근도 병행해가면서 한다.
2 : 헐 정말 고생한다. 그 정도로 하면 난 그 일 못할 것 같다.
1 : 책임감 때문에 어쩔 수 없이 그 일 마무리 하려고 노력한다...
2 : 너의 책임감 때문에 뒷선에서 사람 더 안 뽑아 준다고 생각 안 해봤어?
1 : 아는데. 그게 내 양심상이라고 해야 하나 내 체질상 일 미뤄두고 이걸 안되더라.
1 : 어떻게든 내가 야근을 해서라든지 특근을 해서라든지 해결하려고 해
2 : 아휴 대단하다 대단해!
2 : 네가 책임감 큰 건 알겠는데 그렇게 하면 위에서 알아주질 않아 ㅠ

4. 데이터 증강 방법 - EDA

EDA(Easy Data Augmentation)


- 2019년 EMNLP에서 발표한 논문
- 소량의 텍스트 데이터로도 성능을 높이기 위한 간단한 데이터 증강 기법 제안

이름	설명
SR (Synonym Replacement)	단어를 유의어로 바꾸기 (ex: 방 → 객실, 안방)
RI (Random Insertion)	단어를 유의어로 바꾼 후 랜덤 위치에 삽입
RS (Random Swap)	문장 내 단어의 순서 변경
RD (Random Deletion)	일부 단어를 확률적으로 삭제

KoEDA

- 본 프로젝트에서 사용한 데이터 증강 툴
- 영어 기반 EDA를 한국어에 맞게 변형
- EDA의 4가지 기법을 한국어에 맞춰 적용

모델 실험

A solid red horizontal bar spanning the width of the slide, positioned below the title.

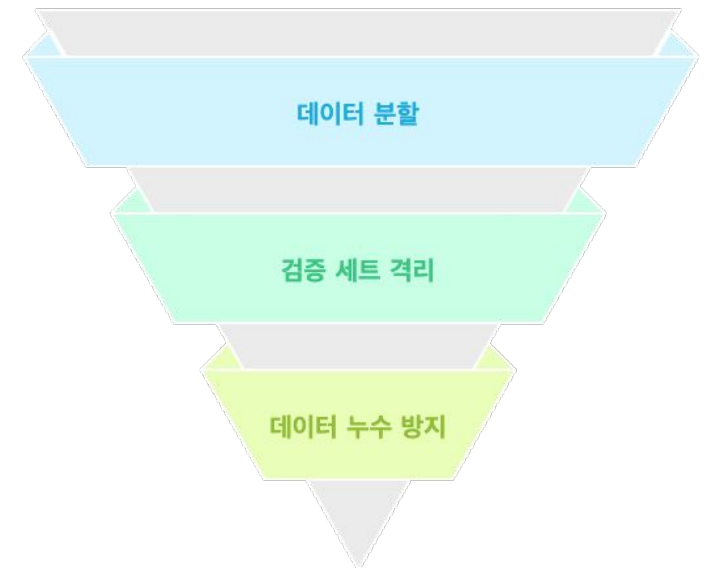
5. 실험 방법

실험 목표

- From-Scratch Model과 Pre-trained Model을 기준으로 실험
 - BiLSTM(64), KcELECTRA, KcBert
- 각 모델에 대해 데이터 증강 여부/일반 데이터 유형에 따른 성능을 비교

평가 기준

- Raw data의 20%를 **Validation set** 으로 저장하여 이를 기준으로 평가
- Validation data를 증강하여 훈련세트에 포함하면 데이터 누수 문제가 발생할 수 있으므로 실험 초기부터 별도로 분리하여 평가 외 목적으로 사용하지 않음
- 검증 세트의 f1-score



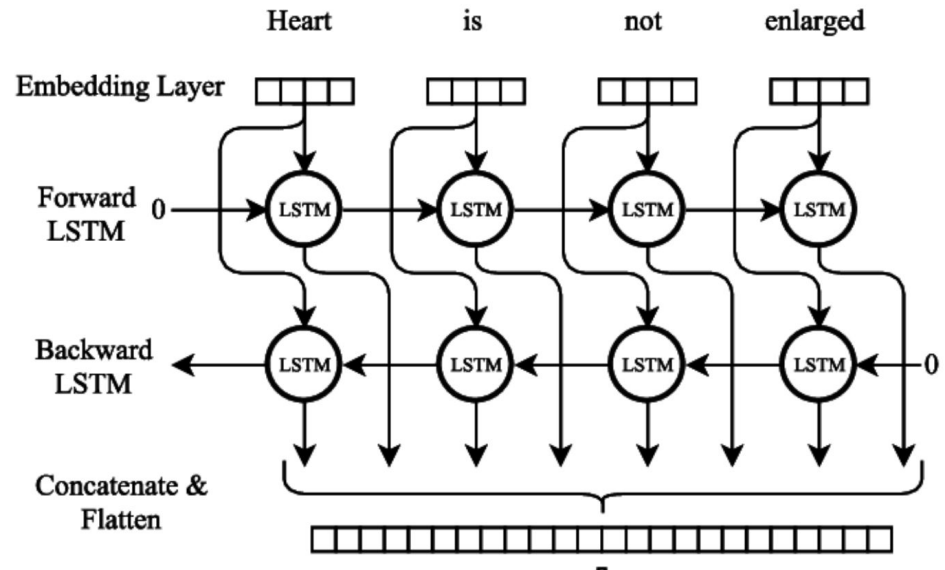
6. Baseline – BiLSTM(64)

양방향 LSTM(Bidirectional LSTM, BiLSTM)

- 양방향 문맥 정보 활용
- 시퀀스 데이터 처리에 널리 사용되는 구조로, 단순하지만 일정 수준 이상의 성능 기대
- 본 프로젝트에선 64개 유닛으로 구성(BiLSTM(64))

Baseline으로 사용한 이유

- 적절한 계산 효율
- 단순하지만 일정 수준 이상의 성능
- 재현성과 범용성이 좋음



6. Baseline – BiLSTM(64)

- Unit: 64개, Embedding_dim = 64, 시퀀스 최대 길이: 350, 어휘 사전 크기: 7000

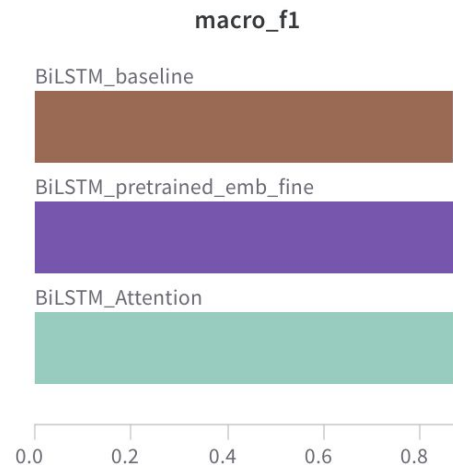
Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 350)	0
embedding (Embedding)	(None, 350, 64)	448,064
bidirectional (Bidirectional)	(None, 128)	66,048
dropout (Dropout)	(None, 128)	0
dense (Dense)	(None, 5)	645

- 랜덤 초기화된 임베딩, 데이터 증강 없음, LLM 생성 일반 데이터 사용
- 데이터 전처리: 구두점 제거, 품사 기준 토큰화
 - Okt를 사용하여 품사(명사, 동사, 형용사, 부사, 숫자)를 기준으로 토큰화 -> 대화의 내용에 집중
- 검증 세트 0.86점, 캐글 리더보드 0.68점
 - EDA 때 예측했던 대로 협박대화와 기타 괴롭힘 대화에 대한 예측 성능이 떨어지는

	precision	recall	f1-score	support
협박 대화	0.83	0.80	0.82	179
갈취 대화	0.85	0.86	0.85	195
직장 내 괴롭힘 대화	0.89	0.87	0.88	194
기타 괴롭힘 대화	0.79	0.81	0.80	202
일반 대화	0.98	1.00	0.99	158
accuracy			0.86	928
macro avg	0.87	0.87	0.87	928
weighted avg	0.86	0.86	0.86	928

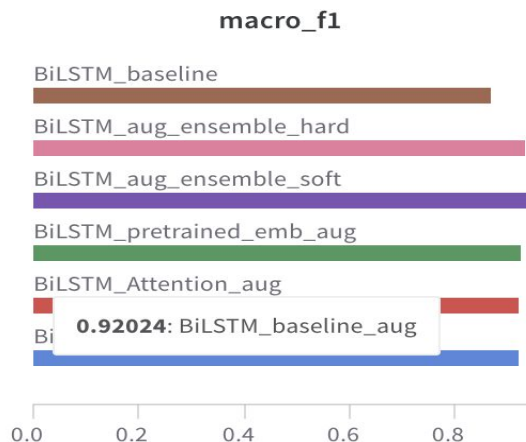
7. From-Scratch Model

- **Baseline** 모델에서 성능을 높이기 위해 하이퍼파라미터를 고정하고 모델 구성요소를 추가/변경
 1. Attention 매커니즘 도입
 2. Pre-trained Embedding 사용(fine-tuning)
- 두 방법에서 모두 검증 세트 점수 향상
 - 일반적으로 이 두 요소는 시퀀스 모델의 성능 개선에 효과적
 - baseline+Attention+Pre-trained Embedding 구조를 From-Scratch Model의 기본 구조로 사용



데이터 증강(KoEDA)에 따른 성능 비교

- Baseline, Attention, Pre-trained Emb 각 3가지 모델에 대해 데이터 증강에 따른 성능 변화를 실험



- 세 모델 모두 증강을 하지 않았을 때 보다 성능 향상 (baseline의 경우 0.86 -> 0.92)
- 세 모델을 앙상블 했을 때 캐글 리더보드 0.71점 기록

7. From-Scratch Model

일반대화 데이터셋에 따른 성능 비교

- baseline에 Attention과 Pre-trained Embedding을 추가한 모델 기준
- 일반대화 데이터셋을 "AIHUB 주제별 텍스트 일상 대화 데이터셋"으로 변경시 성능 측정

결과




- LLM 생성 데이터보다 높은 성능
 - LLM 생성 데이터는 일상대화를 온전히 대표하지 못하는 것으로 예상
- 일반대화 데이터셋의 수를 늘릴수록 일반화 성능 향상
 - 불확실한 테스트셋 일반데이터 클래스에 대해 많은 샘플로 학습해야 일반화된 패턴을 학습할 수 있는 것으로 예상
- 캐글 리더보드 0.74점 기록
 - 기존 검증 세트의 일반 대화 데이터는 LLM 데이터인데 AIHUB 데이터로 학습함에 따라 검증 점수가 저조한 문제 발생
 - 따라서 검증 세트의 일반 대화 데이터를 AIHUB 데이터로 교체하고, 캐글 리더보드를 기준으로만 일반화 성능 평가

8. Pre-trained Model : KcELECTRA

- **모델 구조**
 - BERT 기반 + ELECTRA 방식 (**Discriminator 모델**)
- **학습 방식:**
 - Replaced Token Detection (가짜 단어인지 판별)
 - BERT보다 더 **효율적으로 학습** 가능
- **특징**
 - 적은 연산량으로도 높은 성능
 - 속도 빠르고, 정확도 높음
 - 한국어 버전은 Hugging Face나 SKT에서 공개됨
 - 빠르고 가볍고, 작은 모델로도 성능 잘 나옴
 - 데이터 전처리 방식은 KcBert 와 동일
- **단점**
 - Generator가 없어서 문장 생성에는 부적합

8. Pre-trained Model : KcELECTRA

항목	LLM 일반 대화 셋		AI Hub 일반 대화셋			
	증강 없음	EDA 증강	증강 없음	EDA 증강	EDA 증강 + 일반 추가	EDA 증강 + 일반 추가
일반대화셋 수	891건	2000건 (증강)	1000건	2000건	8000건	16000건
loss	0.55, 0.20	0.49, 0.13	0.46, 0.16	0.50, 0.15	0.41 0.16	0.40 0.15
accuracy	0.90302	0.91595	0.87823	0.89116	0.90517	0.90086
Accuracy f1	0.90648	0.91838	0.88001	0.89291	0.90734	0.90236
일반 대화_f1	0.99054	0.99371	0.91946	0.93069	0.95793	0.9434
갈취_f1	0.86387	0.87437	0.84804	0.86979	0.87113	0.86869
기타 괴롭힘 대화_f1	0.85308	0.88325	0.83582	0.83871	0.86139	0.85496
협박 대화_f1	0.86932	0.87151	0.84507	0.86863	0.8895	0.88333
직장 괴롭힘 대화_f1	0.95561	0.96907	0.95165	0.95674	0.95674	0.96144

	submission_model5_final.csv Complete (after deadline) · TaeHun Ban · 27s ago	0.86012	0.86012	<input type="checkbox"/>
	submission_model2_final (1).csv Complete · TaeHun Ban · 13m ago		0.70977	<input type="checkbox"/>
	submission_model6_final (1).csv Complete · TaeHun Ban · 15m ago		0.85557	<input type="checkbox"/>

8. Pre-trained Model : KcELECTRA

일반 대화(5번째 클래스) 과다 증강 → 성능 하락 원인 분석

1. 문제 현상

- 모델5 (일반대화 8000건) > 모델6 (일반대화 16000건)
- 일반 대화 데이터만 증가했는데 F1 score 하락 발생

2. 원인 분석

- ① 모델 편향
 - 일반 대화 과대표집 → 해당 클래스로 치우친 예측
 - 협박·갈취 등 타 클래스 분류 성능 저하
- ② 증강 노이즈 증가
 - KoEDA 등으로 생성된 문장 중
 - 문맥 애매하거나 품질 낮은 문장 다수 포함 가능성

3. 해결 방향

일반 대화 8000건이 성능 최적점 가능성
16000건 사용할 경우:

- 증강 문장 하드 필터링
- `class_weight` 조정 또는 언더샘플링 적용 고려

9. Pre-trained Model : KcBERT

● 모델 구조

- KoBERT 기반으로 파인튜닝
- base (12- Transformer Encoder layer, 110M parameters)
- 12개 Transformer Encoder Layer (Embedding, Attention, FFN)
- 12개의 Attention Head 수
- Max Sequence Length : 512 토큰
- Embedding : Position + Token + Segment embedding 사용

● 학습방식

- 학습데이터 : 네이버 뉴스 댓글 6,000만 개 이상
- 토큰나이저 : SentencePiece 기반 WordPiece 토큰나이저
- Next Sentence Prediction은 사용하지 않고 MLM (Masked Language Modeling)만 수행

● 특징

- 네이버 뉴스 댓글 데이터에 특화되어 사전학습됨
- 일반적인 BERT보다 일상적이고 구어체적인 한국어 표현에 강함
- 비속어, 줄임말, 감정 표현 등 비정형 한국어에 강함

● 단점

- 학습 데이터가 네이버 뉴스 댓글 위주여서, 법률, 의료, 학술 등 정형적이거나 전문적인 언어 도메인에는 부적합할 수 있음
- 뉴스 댓글 외 환경(예: SNS, 대화체, 긴 문서 요약 등)에서는 성능이 예상보다 낮을 수 있음

9. Pre-trained Model : KcBERT

항목	LLM 일반 대화 셋		AI Hub 일반 대화셋			
	증강 없음	EDA 증강	증강 없음	EDA 증강	EDA 증강 + 일반 추가	EDA 증강 + 일반 추가
일반대화셋 수	891건	2000건 (증강)	1000건	2000건	8000건	16000건
loss	0.45, 0.14, 0.06	0.31, 0.05, 0.03	0.46, 0.14, 0.05	0.2915	0.2023	0.1586
accuracy	0.84, 0.95, 0.98	0.89, 0.98, 0.99	0.84, 0.95, 0.98	0.9035	0.9312	0.9470
Val_loss	0.29, 0.27, 0.35	0.45, 0.43, 0.45	0.30, 0.35, 0.36	0.3118	0.3595	0.3369
Val_accuracy	0.90, 0.91, 0.90	0.86, 0.89, 0.89	0.89, 0.90, 0.89	0.9084	0.8858	0.9019
Accuracy f1	0.17241	0.22522	0.21659	0.19073	0.19181	0.19504
갈취 대화_f1	0.16	0.24818	0.18605	0.18617	0.15467	0.22281
일반 대화_f1	0.19048	0.15385	0.17143	0.1442	0.15576	0.1548
기타 괴롭힘 대화_f1	0.17518	0.21836	0.27184	0.17632	0.25404	0.18421
협박 대화_f1	0.13174	0.24713	0.1813	0.21925	0.1677	0.23158
직장 괴롭힘 대화_f1	0.20202	0.24607	0.25707	0.22051	0.20741	0.17677
macro_f1	0.17188	0.22272	0.21354	0.18929	0.18792	0.19403
Submission 점수	0.67031	0.67484		0.79246		0.83106

- Epoch가 증가시 훈련 데이터의 loss/accuracy는 향상되는데, 검증 데이터의 loss/accuracy는 향상되지 않는 과대적합 현상이 발생하였음
- 데이터 증강시 성능이 향상되었고, LLM 데이터셋보다 AI Hub 데이터셋 사용시 성능이 향상되었음
- Test 데이터에서 '일반 대화' 구별 점수가 낮았으며, 일반대화셋을 늘릴 경우, 성능이 향상되었음

프로젝트 결과

● BiLSTM Model 실험 결과 Insight

Test Case	Test 조건	검증 f1-score	캐글 점수
1. Baseline	데이터 증강없이 LLM 일반대화셋 사용 - Okt 품사기준 토큰화, 단어사전 7,000	0.86	0.68
2. Attention 추가	Baseline에 Attention 추가	0.87	-
3. Embedding Fine tuning	Baseline에 Pre-trained Embedding 사용	0.88	-
4. 데이터 증강	Baseline+Attention+Pre-trained Embedding 모델에 기존 데이터 증강 및 LLM 일반대화셋 사용	0.92	0.71
5. AI Hub 데이터셋	Baseline+Attention+Pre-trained Embedding 모델에 기존 데이터 증강 및 AIHUB 일반대화셋 사용	-	0.74

- I. 적은 데이터셋(약 4천건)의 경우 데이터 증강이 성능 향상에 도움이 되는 것을 확인함
- II. Attention, Pre-trained Embedding 사용이 성능 향상에 도움이 됨
- III. LLM 데이터셋 보다는 AI Hub 데이터셋이 보다 효과적임
- IV. AIHUB 데이터를 사용했을 때 일반대화에 대한 분류 결과가 낮았는데, 일반대화 데이터 수를 늘리는 것이 도움이 되었음

Pre-trained Model 실험 결과 Insight

Test Case	Test 조건	캐글 점수
1. LLM 기본	데이터 증강없이 LLM 일반대화셋 사용	0.67031
2. 데이터 증강	데이터 증강하여 LLM 일반대화셋 사용	0.67484
3. AI Hub 데이터셋	데이터 증강 및 AI Hub 일반대화셋 사용	0.79
4. 일반대화셋 증가	데이터 증강하여 AI Hub 일반대화셋 추가 사용	0.83 ~0.86

- I. Baseline 보다는 pre-trained Bert model이 DKTC 다중 분류를 더 잘함
- II. Baseline Model과 마찬가지로, 데이터 증강 및 AI Hub 일반대화 데이터셋이 성능 향상에 도움을 줌
- III. 괴롭힘, 협박, 갈취 보다 일반 대화 분류 점수는 약하게 나오는데, 비속어나 감정표현등에 특화된 학습 모델이라서 그런 것으로 추측됨
- IV. 일반대화 데이터셋의 개수를 늘렸을 때(비일반대화의 2배) 성능이 좋게 나오는 양상이나, 적절한 일반대화 데이터셋 수 설정이 필요하다
- V. 한국어 모델이라서 영문자 포함된 경우 분류가 안되는 문제가 보임

END

