

# DKTC MULTICLASSIFICATION

---

DLTON AIFFEL ONLINE 5기 NLP

일등 이리5조

# Contents

## Intro

- 0 팀원 소개
- 1 데이터 소개

## EDA

- 2 데이터 전처리 - 텍스트 정제, 토큰화, 불용어 제거 등
- 3 데이터 탐색 및 분석 - 데이터 탐색을 통한 각 클래스 분포
- 4 데이터 분할

## 실험

- 5 토큰나이징, 패딩 - 텍스트 데이터 토큰화: konlpy mecab, sentence piece - 패딩
- 6 모델구성 및 평가

## 결론

- 7 결론
- 8 자가 평가

# 00 팀원 소개

Intro

## 최한준 TMI

- 1 ENTP
- 2 주영님보다 조금 작음
- 3 배드민턴 좋아함
- 4 GIST 대학원생(예정)
- 5 이번주 금요일 개강

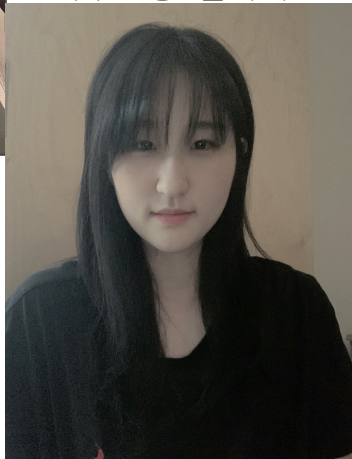


## 홍서이 TMI

- 1 ISFJ
- 2 이번주 금요일 개강
- 3 방콕생활중

## 양주영 TMI

- 1 키 큼(176cm)
- 2 당달에 라섹 예정
- 3 밥 메이트: 이세계 삼촌
- 4 유진님이랑 중딩동창
- 5 복수전공: 철학과



## 신유진 TMI

- 1 키 작음(172cm)
- 2 주영님이랑 중딩동창
- 3 토마토 못먹음
- 4 폰게임3개이상함
- 5 릿앤모티를 시청중

# 01 데이터 소개

Intro

클래스	Class No.	# Training	# Test
협박	00	896	100
갈취	01	981	100
직장 내 괴롭힘	02	979	100
기타 괴롭힘	03	1094	100
일반	04	-	100

idx		class	conversation
0	0	협박 대화	지금 너 스스로를 죽여달라고 애원하는 것인가?\n아닙니다. 죄송합니다.\n죽을 ...
1	1	협박 대화	길동경찰서입니다.\n9시 40분 마트에 폭발물을 설치할거다.\n네?\n꼭바로 들어 ...
2	2	기타 괴롭힘 대화	너 되게 귀여운거 알지? 나보다 작은 남자는 침뱉어.\n그만해. 니들 놀리는거 재미...
3	3	갈취 대화	어이 거기\n네??\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 ...

```
text
t_495  미나씨 휴가 결재 올리기 전에 저랑 상의하라고 말한거 기억해요? 네 합니다. 보고서...
t_496  교수님 제 논문에 제 이름이 없나요? 아 무슨 논문말이야? 지난 번 냈던 논문이...
t_497  야 너 네 저요? 그래 너 왜요 돈좀 줘봐 돈 없어요 돈이 왜 없어 지갑은 품이...
t_498  야 너 빨리 안 뛰어와? 너 이 환자 제대로 봤어 안 봤어 어제 저녁부터 계속 보다...
t_499  엄마 저 그 돈 안해주시면 정말 큰일나요. 이유도 말하지 않고. 몇번패니 경민아....

len(test_data)

400
```

- Input값: 문장
  - Output값: 4개의 클래스로 분류
- => “다중클래스분류”를 해야함

데이터 분포를 확인해보자!

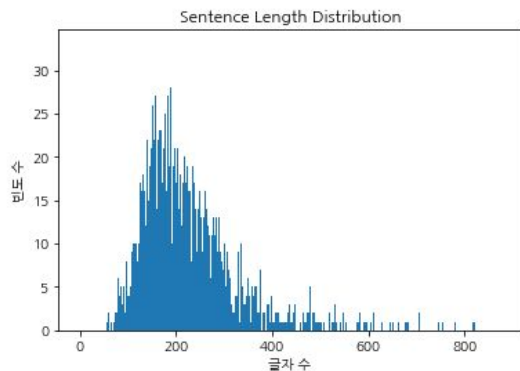
## 02 데이터 탐색 및 분석

EDA

	중복제거
train dataset	3900 -> <b>3846</b>
test dataset	<b>400</b> -> 399

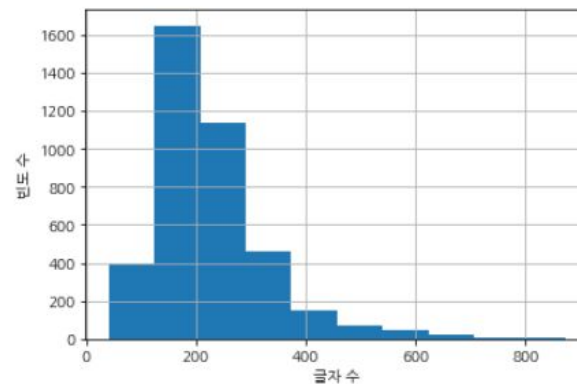
train\_data 중복제거 O

Data Size: 3846  
문장의 최단 길이: 41  
문장의 최장 길이: 874  
문장의 평균 길이: 227



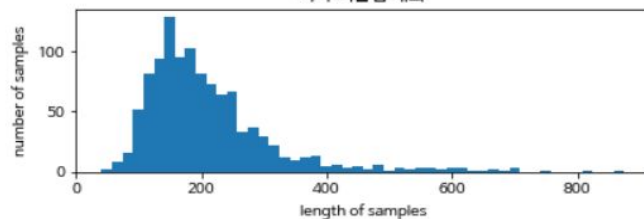
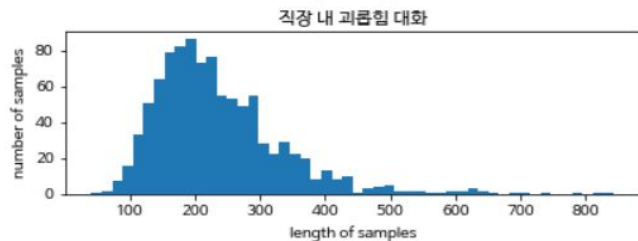
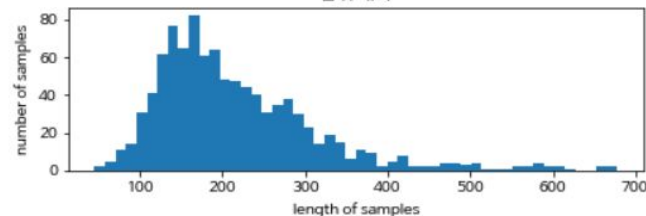
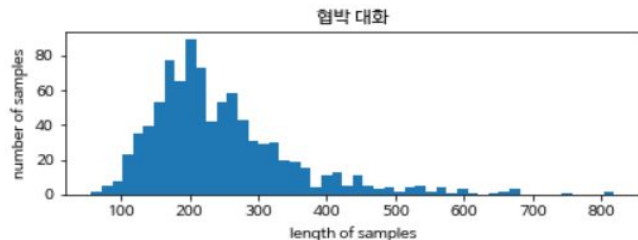
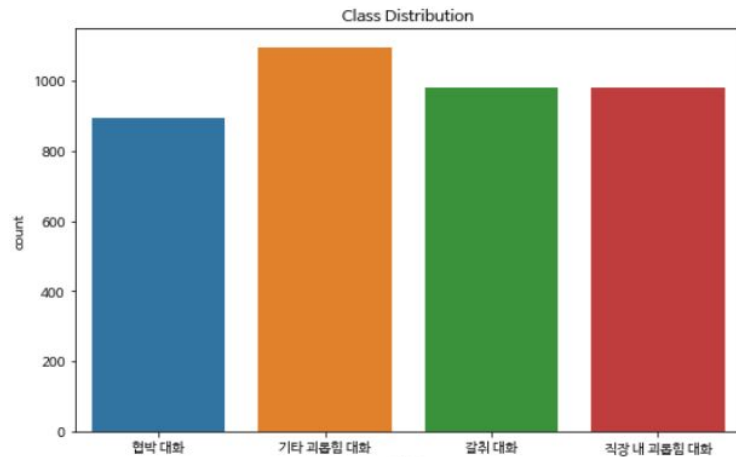
train\_data 중복제거 X

최대길이: 874  
최소길이: 41  
평균길이: 226.57088607594937



# 각 클래스별 데이터 분포도

	협박	갈취	직장	기타
최대길이	818	678	843	874
평균길이	246.073	216.186	237.558	210.079



## 03 데이터 분석 결론

중간점검

관찰한 점	결론
전체 텍스트 길이가 평균 <b>227</b> 로 길다.	장기 기억 손실 고려
<b>train dataset</b> 의 총 개수가 <b>3846</b> 개이다.	단순한 모델
데이터 카테고리 분포가 일정하다.	-불균형 클래스 처리 작업 필요 없음 -통계적으로 모델의 최소 성능이 <b>0.25</b> 이상이어야 함

## 04 데이터 전처리

EDA

	MeCab	Okt	SentencePiece
토큰화	O	O	O
불용어 제거	O	O	O
특징	!!!와 !x8개를 다른 토큰으로 인식	명사를 중심으로 형태소 분석을 수행해 명사 추출에 강점을 가짐	train data로 학습 시 성능이 70%가 나왔다, SentencePiece토큰을 기반으로 하는 BERT를 고려해봄.



# 실험

---

모델 선택 및 실험

# 05 모델구성 및 평가

실험

모델명	Tokenizer	모델컴파일	하이퍼파라미터 (배치 크기, 에포크 수)	평가지표 설정	test accuracy
1D CNN	Mecab	Adam	batch_size = 32, epochs = 10	accuracy	0.26
LSTM					0.25
GRU					0.25
BiLSTM					0.26
LSTM	Mecab, Okt	Adam	그리드 서치 이용	accuracy (confusion metrics를 다 봄)	0.26
LSTM	Sentencepiece	Adam	Callback 함수 사용	accuracy	0.715
klue-BERT (fine-tuned-klue-bert-base)	BertTokenizerFast (SentencePiece 기반)	Adam	Callback 함수 사용 (EarlyStopping, LearningRateScheduler, ModelCheckpoint)	accuracy	<b>0.8975</b>
		RectifiedAdam	RectifiedAdam 코드 그대로 사용	accuracy	<b>0.9125</b>

# 06 모델구성 - LSTM

실험

- **name: LSTM**

- 데이터셋이 문장 형태이고 시퀀스적 특성이 중요하기 때문에, 문맥 정보를 잘 캡처하는 LSTM 모델을 구축
- 모델의 일반화 성능을 높이기 위해 `train_test_split()`에서 **stratify** 옵션 사용
- 그리드 서치를 활용해 최적의 하이퍼파라미터 선택

- **tokenizer: mecab**

- konlpy의 형태소분석기 중 성능이 준수하고 빠른 형태소분석기인 **mecab**이용해 토큰화 진행
- {'embedding\_dim': 100, 'hidden\_units': 256, 'lstm\_units': 128}
- test Accuracy: 0.26

- **tokenizer: sentencepiece**

- 단어가 아니라 **subword**를 생성하여 토큰으로 사용하기에 OOV문제에 강한 **sentencepiece**로 토큰화 진행
- 제공된 데이터 중 'train.txt'를 사용하여 **sentencepiece**를 학습시킴
- {'embedding\_dim': 128, 'hidden\_units': 128, 'lstm\_units': 64}
- test Accuracy: 0.715

## 06 모델구성 - klue-BERT-classification

실험

- **name : "klue/bert-base"**

BERT를 사용한 이유는 BERT는 Text를 Encoding을 하며 정보를 추출하는게 뛰어난 모델이며, 이를 활용해 DownStream Task에 적용하기 적합한 모델이라 판단했습니다.

- model : TFBertForSequenceClassification

- **tokenizer : BertTokenizerFast**

기존 pretrained tokenizer에서는 '\n'이 없는데, 본 데이터에서 '\n'의 경우 “ 발화자를 나누는 중요한 word라 생각되어 tokenizer에 vocab에 추가했습니다.

- EarlyStopping, LearningRateScheduler, ModelCheckpoint와 같은 Callback 함수를 사용해 **학습 과정을 안정화**했습니다.

```
import tensorflow_addons as tfa
```

- ```
optimizer = tfa.optimizers.RectifiedAdam(lr=5.0e-5, total_steps = 2344*3, warmup_proportion=0.1, min_lr=1e-5, epsilon=1e-08, clipnorm=1.0)
model.compile(optimizer=optimizer, loss=model.compute_loss, metrics=['accuracy'])
```

# 결론

---

결론 및 Self-Check

## <진행하면서 배운점>







- 리더보드 스코어 0.9125 달성
- 자연어처리 워크플로우에 대한 이해도 증가
- pretrained model 처음으로 사용해봄
- 문장부호 처리 여부에 따라 토큰화가 달라지는 점
- 토큰화 방식에 따라 성능이 달라지는 점
- 사전 훈련된 모델에서 성능이 비교적 좋았던 점

## <아쉬웠던 점/ 더 시도해보고 싶은 점>

- 문장 기호가 포함된 토큰화를 시도해보고 싶은 점
- 다양한 파라미터 튜닝을 시도해도 0.9125를 넘지 못해 아쉬웠던 점.
- 다른 사전 훈련된 모델을 시도해보고 싶은 점
- 앙상블을 시도해보고 싶은 점
- 데이터 증강을 시도해보고 싶은 점

## 08 Self-Check

결론

- 데이터 **EDA**와 데이터 전처리가 적절하게 이루어졌는가? 
- **Task**에 알맞게 적절한 모델을 찾아보고 선정했는가? 
- 성능향상을 위해 논리적으로 접근했는가? 
- 결과 도출을 위해 여러가지 시도를 진행했는가? 
- 도출된 결론에 충분한 설득력이 있는가? 
- 적절한 **metric**을 설정하고 그 사용 근거 및 결과를 분석하였는가? 

# 09 Reference

결론

- [KLUE: Korean Language Understanding Evaluation](#)
- [Long Short-Term Memory](#)
- <https://parksrazor.tistory.com/231>
- [KLUE-BERT github](#)