ZeroToOne

Start

DLThon

Haewon Park, Gyubin Hwang, Yeonsoo Kim, Bumjun Kim

Task: TEXT CLASSIFICATION

OVERVIEW

01
APPROACH
Keywords

02
DATA
Exploring and Analyzing data for EDA

03
MACHINE LEARNING - PREPROCESSING

04
COMPARISON
ML vs Pre-trained model

05
PRE-TRAINED MODELS
Reason for selection

06
INSIGHT
insufficient data

07
DATA AUGMENTATION
KorEDA
AI Hub data
Back Translation

08
RESULT
skt/KoGPT2-base-v2
KcELECTRO-small
Klue/bert-base

APPROACH

평가 기준

1. 데이터 EDA와 데이터 전처리가 적절하게 이뤄졌는가?
2. task에 알맞게 적잘한 모델을 찾아보고 선정했는가?
3. 성능향상을 위해 논리적으로 접근했는가?
4. 결과 도출을 위해 여러 가지 시도를 진행했는가?
5. 도출된 결론에 충분한 설득력이 있는가?
6. 적절한 metric을 설정하고 그 사용 근거 및 결과를 분석하였는가?
7. 발표가 매끄럽게 진행되었고 발표시간을 준수하였는가?

KEYWORDS

EDA (Exploratory Data Analysis)
Preprocessing
Model
Performance
Metrics

INITIAL ACTION

Directly Exploring and Analyzing Data for EDA

DATA

Directly Exploring and Analyzing Data for EDA

2,기타 괴롭힘 대화,"너 되게 귀여운거 알지? 나보다 작은 남자는 첨봤어.
그만해. 니들 놀리는거 재미없어.
지영아 너가 키 160이지? 그럼 재는 160도 안돼는거네?
너 군대도 안가고 좋겠다.
니들이 나 작은데 보태준거 있냐?
난쟁이들도 장가가고하던데. 너도 희망을 가져봐
더이상 하지마라.
그 키크는 수술도 있대잖아? 니네 엄마는 그거 안해주디?
나람 해줬어. 저 키로 어찌살아.
제발 그만 괴롭히라고!"
3,갈취 대화,"어이 거기
예??
너 말이야 너. 이리 오라고
무슨 일.
너 옷 좋아보인다?
얘 돈 좀 있나봐
아니에요.돈 없어요
뒤져서 나오면 넌 죽는다
오늘 피시방 콜?
콜. 마지막 기회다. 있는거 다 내놔
정말 없어요"

t_001/text	2	우리팀에서 다른팀으로 갈 사람 없나? 그럼 영지씨가 가는건 어때? 네? 제가요? 그렇지? 2달만 파견 잘 갔다오면 승진이야. 네? 저는 별로 가고 싶지 않습니다. 여기 있는 모든사람도 가기 싫어해. 그
t_002/text	3	너 오늘 그게 뭐야 네 제가 뭘 잘못했나요.? 제대로 좀 하지 네 똑바로 좀 하지 행실이 맘에 안들어 네 실천할께요 옆에 동기들 좀 본받으란 말야 어? 네 알겠습니다 그래 똑바로해 노력할께요
t_003/text	4	이거 들어바 와 이 노래 진짜 좋다 그치 요즘 이 것만 들어 진짜 너무 좋다 내가 요즘 듣는 것도 들어봐 음 난 좀 별론데 좋을 줄 알았는데 아쉽네 내 취향은 아닌 듯 배고프다 밥이나 먹으러 가자 그래
t_004/text	3	아무튼 앞으로 니가 내 와이파이야. .응 와이파이 온. 켰어. 반말? 주인님이라고도 말해야지? .켰습니다. 주인님. 오냐. 앞으로도 잘부탁한다 와이파이. 넵. 이렇게만 해? 알겠지? 응.
t_005/text	0	그러니까 빨리 말해. 선생님 제발 살려주십시오. 비밀번호 틀릴 때마다 손톱 하나씩 뺀찌로 뽑는다. 선생님 저도 정말 모릅니다. 하나 엄지 손톱 뽑는다. 으악! 잘못했습니다. 그런데 진짜 모릅니다.
t_006/text	0	그 사람 누구냐? 누구말하는거야? 다 알면서 모른척 하지마. 둘다 쏴버리기 전에. 아니 정말 왜이래? 누굴말하는거냐고. 어제 밤에 전화온 그새끼 말이야. 번호 알아놨으니까 죽이기 전에 다 불어라.
t_007/text	1	야 저기야 닭꼬치 트럭왔다 응 그러네 그치? 너도 먹고 싶지? 어?나도? 그래 너 세상에 닭꼬치 싫어하는 사람이 어딨어 그렇긴한데 배부른데 그래? 그럼 나만 먹고 올게 응. 뭐해?? 뭘? 돈줘야 먹고오

MACHINE LEARNING – PREPROCESSING

```
train_data = train_data.sample(frac = 1, random_state = 42)

def preprocess_sentence(sentence):

    sentence = sentence.lower().strip()
    sentence = re.sub(r"([?.!,])", r" \1 ", sentence)
    sentence = re.sub(r'[" "]+', " ", sentence)
    sentence = re.sub(r"[^a-zA-Z?.!가-힣ㄱ-ㅎㅏ-ㅣ]+", " ", sentence)
    sentence = sentence.strip()

    return sentence
```

1. Transform lowercase letters, remove spaces

2. . ? ! Handles spaces before and after punctuation marks such as , etc.

3. If there are two or more spaces, one space is processed.

4. Remove characters other than a~z, A~Z, ?, ., !, 가~힣, ㄱ~ㅎ, ㅏ~ㅣ, etc.

5. remove spaces

MACHINE LEARNING – PREPROCESSING

```
def check_class(it):
    if '협박' in it:
        return 0
    elif '갈취' in it:
        return 1
    elif '직장 내 괴롭힘' in it:
        return 2
    elif '기타 괴롭힘' in it:
        return 3
```

The four classes, including intimidation, extortion, workplace harassment, and other harassment, were assigned 0, 1, 2, and 3, respectively.

COMPARISON

ML VS KLUE/BERT-BASE

```
Linear Support Vector Machine Accuracy: 0.8291139240506329
Logistic Regression Accuracy: 0.8063291139240506
Decision Trees Accuracy: 0.6354430379746835
Random Forest Accuracy: 0.7556962025316456
K-Nearest Neighbors Accuracy: 0.7240506329113924
Naive Bayes Accuracy: 0.8354430379746836
Gradient Boosting Accuracy: 0.7506329113924051
Linear Discriminant Analysis Accuracy: 0.46455696202531643
```

LLM Pretrained Models	klue-bert 5 epoch	klue-bert 3 epoch	klue-bert 4 epoch
Data Original	val_loss: 0.4654 - val_accuracy: 0.8962 ACCURACY : 0.9	val_loss: 0.4078 - val_accuracy: 0.8886 ACCURACY : 0.9025	al_loss: 0.4928 - val_accuracy: 0.8797 ACCURACY: 0.895

The Naive Bayes model, exhibiting the highest accuracy, achieves an approximate accuracy rate of 83.5%, whereas the Klue/bert-base model attains a higher accuracy at around 90%.

PRE-TRAINED MODELS

Baseline model: Klue/bert-base

Trial:

- skt/kogpt2-base-v2
- beomi/KcELECTRA-small-v2022

REASON FOR SELECTION

Klue/bert-base: 다양한 데이터를 사전학습한 모델이므로 전반적으로 준수한 성능을 보일 수 있을 것 같아 베이스라인으로 사용하게 되었습니다.

skt/kogpt2-base-v2: 학습이 빠르고, 인코더 + 디코더 구조의 모델 성능을 비교하고자 사용

beomi/KcELECTRA-small

PRE-TRAINED MODELS

REASON FOR SELECTION

beomi/KcELECTRA-small:

	Size (용량)	NSMC (acc)	Naver NER (F1)	PAWS (acc)	KorNLI (acc)	KorSTS (spearman)	Question Pair (acc)	KorQuaD (Dev) (EM/F1)
KcELECTRA-base-v2022	475M	**91.97**	87.35	76.50	82.12	83.67	95.12	69.00 / 90.40
KcELECTRA-base	475M	91.71	86.90	74.80	81.65	82.65	**95.78**	70.60 / 90.11
KcBERT-Base	417M	89.62	84.34	66.95	74.85	75.57	93.93	60.25 / 84.39
KcBERT-Large	1.2G	90.68	85.53	70.15	76.99	77.49	94.06	62.16 / 86.64
KoBERT	351M	89.63	86.11	80.65	79.00	79.64	93.93	52.81 / 80.27
XLM-Roberta-Base	1.03G	89.49	86.26	82.95	79.92	79.09	93.53	64.70 / 88.94
HanBERT	614M	90.16	87.31	82.40	80.89	83.33	94.19	78.74 / 92.02

NSMC (acc) – Naver Sentiment Movie Corpus (Accuracy)

The NSMC task involves sentiment analysis for movie reviews. The goal is to determine the sentiment expressed in a movie review as either positive or negative. The evaluation metric is accuracy, which measures the proportion of correctly classified sentiments.

INSUFFICIENT DATA

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

DATA AUGMENTATION

KorEDA

AI Hub Data

Back Translation

KorEDA

이 프로젝트는 EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks 를 한국어로 쓸 수 있도록 wordnet 부분만 교체한 프로젝트 입니다.

wordnet은 KAIST에서 만든 Korean WordNet(KWN) 을 사용했습니다.

github.com/catSirub/KorEDA

```
def EDA(sentence, alpha_sr=0.1, alpha_ri=0.1, alpha_rs=0.1, p_rd=0.1, num_aug=9):
```

```
def EDA(sentence, alpha_sr=0.1, alpha_ri=0.1, alpha_rs=0.1, p_rd=0.1, num_aug=9):
        sentence = get_only_hangul(sentence)
        words = sentence.split(' ')
        words = [word for word in words if word is not ""]
        num_words = len(words)

        augmented_sentences = []
        num_new_per_technique = int(num_aug/4) + 1

        n_sr = max(1, int(alpha_sr*num_words))
        n_ri = max(1, int(alpha_ri*num_words))
        n_rs = max(1, int(alpha_rs*num_words))

        # sr
        for _ in range(num_new_per_technique):
                a_words = synonym_replacement(words, n_sr)
                augmented_sentences.append(' '.join(a_words))

        # ri
        for _ in range(num_new_per_technique):
                a_words = random_insertion(words, n_ri)
                augmented_sentences.append(' '.join(a_words))

        # rs
        for _ in range(num_new_per_technique):
                a_words = random_swap(words, n_rs)
                augmented_sentences.append(" ".join(a_words))

        # rd
        for _ in range(num_new_per_technique):
                a_words = random_deletion(words, p_rd)
                augmented_sentences.append(" ".join(a_words))
```

KorEDA – Hyperparameters

sr
synonym replacement

ri
random insertion

rs
random swap

rd
random deletion

https://github.com/jasonwei20/eda_nlp

- **Synonym Replacement (SR):** Randomly choose *n* words from the sentence that are not stop words. Replace each of these words with one of its synonyms chosen at random.
- **Random Insertion (RI):** Find a random synonym of a random word in the sentence that is not a stop word. Insert that synonym into a random position in the sentence. Do this *n* times.
- **Random Swap (RS):** Randomly choose two words in the sentence and swap their positions. Do this *n* times.
- **Random Deletion (RD):** For each word in the sentence, randomly remove it with probability *p*.

sr
synonym replacement

ri
random insertion

rs
random swap

rd
random deletion

원문 데이터

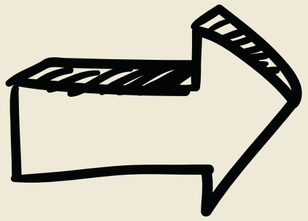
제가 우울감을 느낀지는 오래됐는데 점점 개선되고 있다고 느껴요

data augmentation한 데이터

우울감을 느낀지는 오래됐는데 점점 개선되고 있다고
제가 우울감을 느낀지는 오래됐는데 느껴요 개선되고 있다고 점점
오래됐는데 우울감을 느낀지는 제가 점점 개선되고 있다고 느껴요
느껴요 우울감을 느낀지는 오래됐는데 점점 개선되고 있다고 제가

Example 1: 씨 이번에 또 틀렸잖아 . 죄송합니다 . 다시 확인하겠습니다 . 동기인 씨는 한번 알려주면 끝인데 자네는 매번 알려줘도 이래 . 죄송하지만 팀장님한테 확인받고 한 작업입니다 . 뭐 그럼 내가 일부러 그런다는 거야 . 그건 아니지만 . 그런 식으로 일할거면 회 관둬요 . 누가 보면 내가 괴롭힌다고 생각하겠어 그런 말씀은 지나치십니다 . 그럼 알아서 잘 하던가 . 눈치도 없이 . 다음부터는 다시 확인하겠습니다 . 팀장님도 말씀 가려서 해주세요 .
Example 2: 씨 이번에 또 틀렸잖아 . 죄송합니다 . 다시 확인하겠습니다 . 동기인 씨는 한번 알려주면 끝인데 자네는 매번 이래 . 죄송하지만 팀장님한테 확인받고 한 작업입니다 . 뭐 그럼 내가 일부러 거야 . 그건 . 그런 식으로 일할거면 회사 관둬요 . 누가 보면 내가 괴롭힌다고 생각하겠어 그런 말씀은 지나치십니다 그럼 알아서 잘 하던가 . 눈치도 . 다음부터는 다시 확인하겠습니다 . 팀장님도 말씀 가려서 .
Example 3: 씨 이번에 또 틀렸잖아 . 죄송합니다 . 다시 확인하겠습니다 . 동기인 씨는 한번 알려주면 끝인데 자네는 매번 알려줘도 이래 . 죄송하지만 팀장님한테 확인받고 한 작업입니다 . 뭐 그럼 내가 일부러 그런다는 거야 . 그건 아니지만 . 그런 식으로 일할거면 회 관둬요 . 누가 보면 내가 괴롭힌다고 생각하겠어 그런 말씀은 지나치십니다 . 그럼 알아서 잘 하던가 . 눈치도 없이 . 다음부터는 다시 확인하겠습니다 . 팀장님도 말씀 가려서 해주세요 .
Example 4: 씨 이번에 또 틀렸잖아 . 죄송합니다 . 다시 확인하겠습니다 . 동기인 씨는 한번 알려주면 끝인데 매번 알려줘도 죄송하지만 팀장님한테 확인받고 한 작업입니다 . 뭐 그럼 내가 일부러 그런다는 거야 그건 아니지만 . 그런 식으로 일할거면 회사 관둬요 누가 보면 내가 괴롭힌다고 생각하겠어 그런 말씀은 지나치십니다 . 알아서 잘 하던가 . 눈치도 없이 . 다음부터는 다시 확인하겠습니다 . 팀장님도 가려서 해주세요 .
Example 5: 씨 씨 씨 씨 씨 씨 씨 씨 이번에 회 씨 또 씨 회 씨 씨 씨 틀렸잖아 회 씨 씨 씨 씨 씨 . 씨 씨 죄송합니다 회 씨 . 씨 씨 씨 씨 회 씨 회 다시 확인하겠습니다 씨 씨 씨 . 씨 동기인 씨 씨 씨는 회 한번 씨 알려주면 회 씨 씨 씨 씨 회 씨 끝인데 씨 자네는 회 씨 씨 씨 씨 씨 씨 씨 매번 씨 씨 씨 회 회 씨 씨 씨 씨 회 회 씨 씨 회 씨 씨 알려줘도 씨 회 씨 씨 씨 이래 씨 씨 씨 씨 씨 씨 . 씨 회 씨 죄송하지만 씨 씨 씨 회 회 씨 씨 회 씨 씨 씨 씨 씨 씨 씨 회 씨 씨 씨 씨 회 씨 씨 회 씨 씨 씨 씨 팀장님한테 씨 확인받고 회 씨 씨 씨 씨 회 씨 씨 씨 씨 회 씨 회 회 회 씨 회 씨 한 작업입니다 회 . 회 씨 씨 씨 뭐 회 씨 씨 그럼 씨 회 씨 씨 내가 씨 회 일부러 씨 씨 씨 회 씨 그런다는 거야 씨 씨 씨 씨 씨 회 씨 . 회 회 그건 회 아니지만 회 . 씨 씨 씨 씨 회 씨 회 씨 씨 씨 그런 씨 회 식으로 일할거면 씨 회 회사 씨 회 씨 씨 회 씨 씨 관둬요 씨 씨 씨 씨 씨 씨 씨 씨 씨 씨 씨 회 . 씨 씨 씨 누가 씨 씨 회 보면 씨 회 씨 내가 씨 씨 씨 회 씨 씨 씨 괴롭힌다고 씨 생각하겠어 회 씨 씨 씨 씨 그런 회 씨 씨 씨 회 말씀은 지나치십니다 씨 회 씨 씨 회 씨 씨 씨 씨 회 씨 씨 씨 씨 씨 씨 씨 씨 씨 씨 씨 씨 . 그럼 씨 씨 회 씨 씨 씨 알아서 씨 잘 회 씨 회 하던가 . 눈치도 씨 씨 회 씨 회 없이 씨 씨 씨 . 회 씨 씨 회 씨 회 다음부터는 회 씨 회 씨 씨 씨 씨 씨 회 회 회 회 회 씨 회 다시 회 씨 씨 회 씨 확인하겠습니다 씨 씨 회 회 씨 씨 씨 씨 씨 회 회 회 씨 씨 회 씨 씨 씨 회 씨 씨 . 회 씨 회 씨 씨 팀장님도 씨 씨 씨 씨 씨 씨 회 씨 회 말씀 씨 씨 씨 씨 씨 씨 씨 가려서 씨 회 해주세요 씨 씨 .

AI Hub Data



대화체의 짧은 데이터로 구성

원본 데이터

```
26,협박,니 자식 야채 먹으면 넌 내 손에 죽는다.
90,협박,(()) 먹는다는 표현도 하지 마 혼나요.
102,협박,나 이렇게 되면 나 이제 나 &name4&이 형 무조건 이제 견제야
187,협박,&name1& 시험 볼 때 제발 뒤 좀 돌아보지 마. 다음에 또 걸리면 그땐 진짜 빵점이에요.
189,협박,앞으로 저 &party-name1& 의원들은 골프공을 쳐서 멀리 날아가는 만큼 국민들 지지도 멀리 날아가는 거여 이렇게 생각하라고
195,협박,여러분들이 그 사회에 지금 그래도 아 브레인이라고 소위 얘기하는 지식인들인데 (()) 못 맞히면 안 됩니다.
196,협박,&name4& 원장님 만약에 오늘 문제 한 문제도 못 맞추면 저희가 환자들 빼돌리겠습니다.
343,협박,죽는다
```

전처리 데이터

```
26,협박,니 자식 야채 먹으면 넌 내 손에 죽는다.
90,협박,먹는다는 표현도 하지 마 혼나요.
102,협박,나 이렇게 되면 나 이제 나 이 형 무조건 이제 견제야
187,협박,시험 볼 때 제발 뒤 좀 돌아보지 마. 다음에 또 걸리면 그땐 진짜 빵점이에요.
189,협박,앞으로 저 의원들은 골프공을 쳐서 멀리 날아가는 만큼 국민들 지지도 멀리 날아가는 거여 이렇게 생각하라고
195,협박,여러분들이 그 사회에 지금 그래도 아 브레인이라고 소위 얘기하는 지식인들인데 못 맞히면 안 됩니다.
196,협박,원장님 만약에 오늘 문제 한 문제도 못 맞추면 저희가 환자들 빼돌리겠습니다.
```

JSON File

PARSING

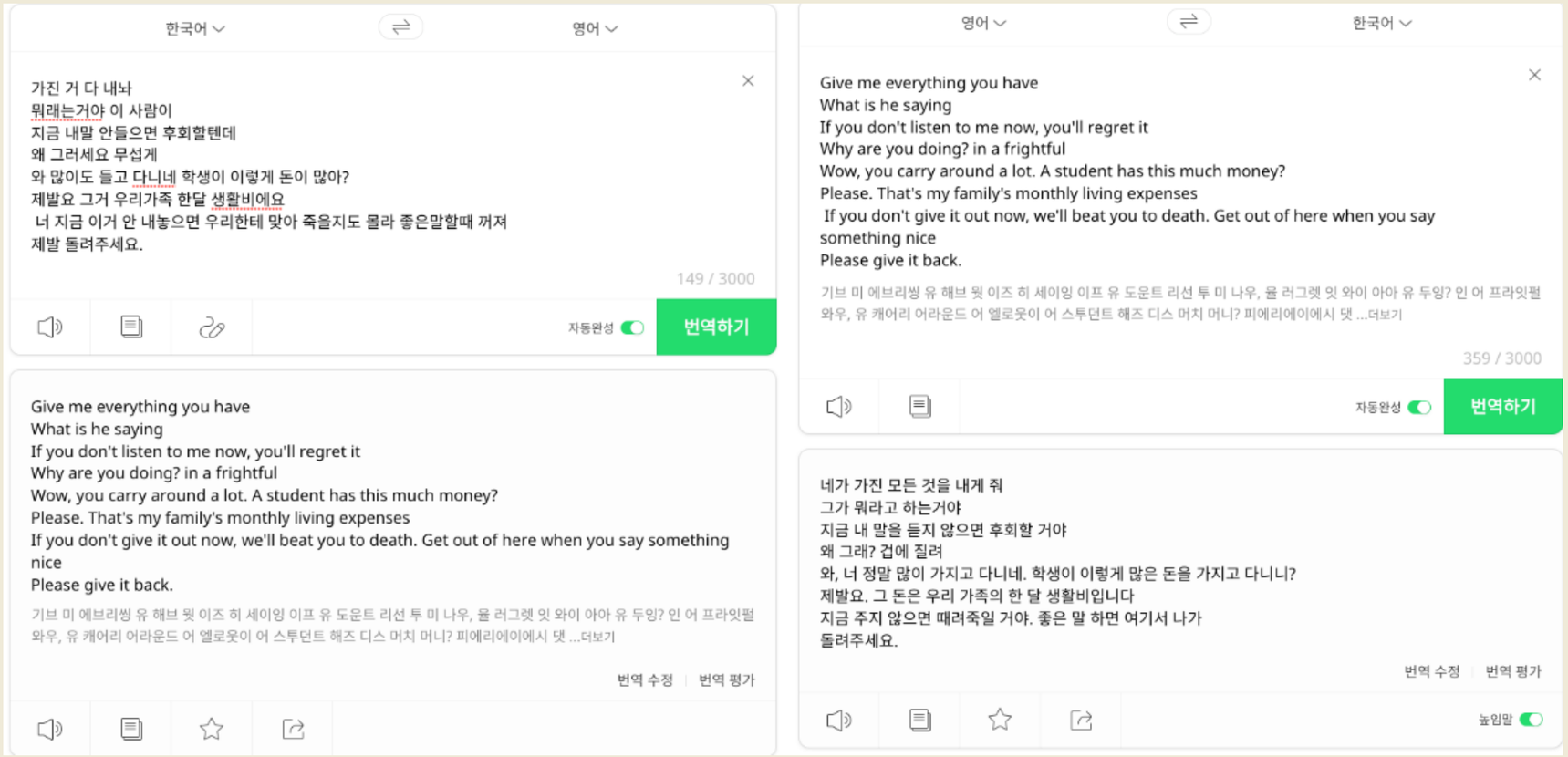

```
{
  "metadata": {
    "title": "A220001",
    "creator": "솔트룩스",
    "distributor": "솔트룩스",
    "year": "2022",
    "group": "구어 > 공적 > 방송",
    "date": "20210921",
    "media": "라디오",
    "publisher": "TBS",
    "category": "교양",
    "program": "경제발전소 박연미입니다",
    "speaker_num": 2,
    "annotation_level": "원시",
    "sampling": "본문 전체"
  },
  "environment": "배경음악, 적음",
  "utterance": [
    {
      "id": "A220001.1.1.1",
      "speaker_id": "0001",
      "start": 0.000,
      "end": 15.550,
      "form": "꽤 긴 연휴인데 직장으로 돌아갈 생각하면 좀 아쉬운 시간이지만 또 투자자들은 고 사이에 한국장이 쉬어가니까 너무너무 손이 간질간질해서 미국 장에 투자를 더 늘린다 이런 얘기도 있더군요.",
      "original_form": "꽤 긴 연휴인데 직장으로 돌아갈 생각하면 좀 아쉬운 시간이지만 또 투자자들은 고 사이에 한국장이 쉬어가니까 너무너무 손이 간질간질해서 미국 장에 투자를 더 늘린다 이런 얘기도 있
      "hangeulToEnglish": null,
      "hangeulToNumber": null,
      "term": null,
      "intent": ["단순 진술"],
      "endpoint": null,
      "summary": null
    }]
}
```


```python
# 폴더 리스트 탐색s
for folder_name in folder_list:
    print("="* 5 ,folder_name)
    folder_path = f"{root_path}/{folder_name}"
    file_list = os.listdir(folder_path)

    # 파일 리스트 탐색
    for file_name in tqdm(file_list):

        if file_name.endswith(".json"):
            file_path = os.path.join(folder_path, file_name)

            with open(file_path, "r") as json_file:
                data = json.load(json_file)

                # filtered_entries = [node for node in data["utterance"] if "협박" in node.get("intent", "")]
                if "utterance" in data and isinstance(data["utterance"], list):

                    for node in data["utterance"]:
                        # Check if "intent" key exists and is a string
                        if "intent" in node and isinstance(node["intent"], list):
                            # Check if "intimidation" is present in the intent
                            if "협박" in node["intent"][0]:
                                txt = node["original_form"]
                                #txt = preprocess_sentence(txt)
                                if len(txt) >= 5 :
                                    data = [idx, "협박", txt]
                                    filtered_entries.append(data)
                                    idx+=1
                else:
                    print("Error: Invalid data structure or missing key 'utterance' in data.")
```

BACK TRANSLATION – PAPAGO

Back Translation: method used to generate a synthetic dataset by first translating textual information into another language and subsequently translating it back into the original language.

Recognizing that the essence of the sentence remains intact even after undergoing the process of back translation, So we concluded that it could be applied to our DLThon task.



BACK TRANSLATION – API TOKEN, HTTP

```
# labels에 id, pw가 리스트 형태로 저장
file_path = '/Users/hwang-gyubin/Downloads/030.방송콘텐츠 대화체 음성인식 데이터/01.데이터/Training/02.라벨링데이터/papago/key.txt'
labels = open(file_path).read().split(",")

client_id = labels[0] # 개발자센터에서 발급받은 Client ID 값
client_secret = labels[1] # 개발자센터에서 발급받은 Client Secret 값
```

```
def translate(text, lang='ko', target_lang='en') :
    encText = urllib.parse.quote(text)
    data = f"source={lang}&target={target_lang}&text=" + encText
    url = "https://openapi.naver.com/v1/papago/n2mt"
    request = urllib.request.Request(url)
    request.add_header("X-Naver-Client-Id",client_id)
    request.add_header("X-Naver-Client-Secret",client_secret)
    response = urllib.request.urlopen(request, data=data.encode("utf-8"))
    rescode = response.getcode()
    if(rescode==200):
        response_body = response.read()
        result  = response_body.decode('utf-8')
        d = json.loads(result)
        print('--- translate --- ')
        print('번역전 : ', text)
        print('번역후 : ',d['message']['result']['translatedText'])
    else:
        print("Error Code:" + rescode)

    return d['message']['result']['translatedText']
```

```
translate("안녕하세요")
```

```
--- translate ---
번역전 :  안녕하세요
번역후 :  Hello

'Hello'
```

규빈님의 수고..

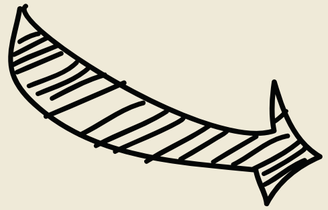

비로그인 오픈 API 당일 사용량
API호출량/일일허용량
Papago 번역
10041/10000
비로그인 오픈 API 당일 사용량
API호출량/일일허용량
Papago 번역
10028/10000
비로그인 오픈 API 당일 사용량
API호출량/일일허용량
Papago 번역
10259/10000

DATA SIZE

기존 데이터

Alhub

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



Alhub
+
back
translation
(파파고)

class	conversation
갈취 대화	981
기타 괴롭힘 대화	1094
직장 내 괴롭힘 대화	979
협박 대화	1239

class	conversation
갈취 대화	1030
기타 괴롭힘 대화	1094
직장 내 괴롭힘 대화	979
협박 대화	1239

RESULT – KCELECTRA-SMALL

LLM Pretrained Models	KcELECTRO - small 2 epoch
Data Original batchsize 16	val_loss: 0.9146 val_F1: 0.81 ACCURACY : 0.815
Data aug + batchsize 16	(A,B) ACCURACY: 0.7975

RESULT – SKT/KOGPT2-BASE-V2

LLM Pretrained Models	skt/kogpt2-base-v2 3 epoch	skt/kogpt2-base-v2 6 epoch	skt/kogpt2-base-v2 9 epoch
Data Original + batchsize 16	val_loss: 0.4956 - val_accuracy: 0.8506 ACCURACY: 0.8625	val_loss: 0.6585 - val_accuracy: 0.8608 ACCURACY: 0.88	val_loss: 0.8069 - val_accuracy: 0.8304 ACCURACY: 0.84
Data aug + batchsize 16	(AI Hub data, Back Translation) ACCURACY: 0.8775	(AI Hub data, Back Translation) ACCURACY: 0.875	

RESULT – KLUE/BERT – BASE

LLM Pretrained Models	klue-bert 3 epoch	klue-bert 4 epoch	klue-bert 5 epoch	klue-bert
Data Original + batchsize 16	val_loss: 0.4078 - val_accuracy: 0.8886 ACCURACY : 0.9025	val_loss: 0.4928 - val_accuracy: 0.8797 ACCURACY: 0.895	val_loss: 0.5031 val_accuracy : 0.8886 ACCURACY: 0.90	
Data without stopwords + batchsize 64	klue-bert 3 epoch val_loss: 0.3629 - val_accuracy: 0.9126 ACCURACY: 0.91	klue-bert 4 epoch val_loss: 0.3791 - val_accuracy: 0.9126 ACCURACY: 0.915	klue-bert 6 epoch val_loss: 0.3742 - val_accuracy: 0.9149 ACCURACY: 0.92	klue-bert 7 epoch val_loss: 0.3944 val_accuracy: 0.9149 ACCURACY: 0.92
Data aug + batchsize 16	(KorEDA) ACCURACY: 0.88 (AI Hub data, Back Translation) val_loss: 0.3467 - val_accuracy: 0.9034 ACCURACY: 0.89	(AI Hub data, Back Translation) val_loss: 0.3134 - val_accuracy: 0.9172 ACCURACY: 0.9	(AI Hub data, Back Translation) val_loss: 0.3826 - val_accuracy: 0.8966 ACCURACY: 0.905	ACCURACY: 0.895

w/o stopwords

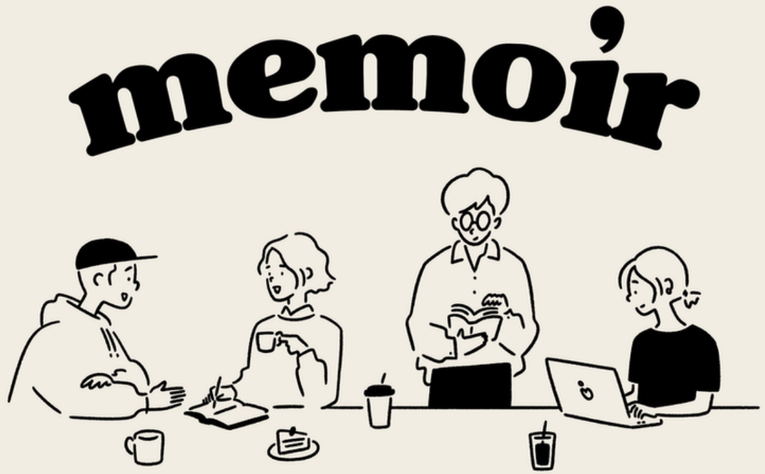
bat size: 64

Accuracy: 0.92

Notebook: https://github.com/dellaanima/DLton_Zero_to_One/blob/main/finetune-hfmodel/

Reference: https://gist.github.com/spikeekips/40eea22ef4a89f629abd87eed535ac6a

RETROSPECT



THANK YOU