BOAZ 분석 23기 미니프로젝트1 2조

텍스트-음성 데이터 기반 멀티모달 감정 분류 모델

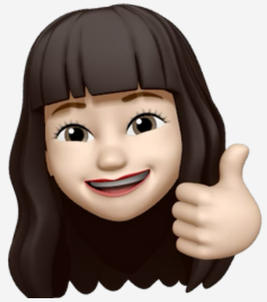
분석 23기 김무연 김윤주 박혜원 백다은 송여경

목차

0. 팀소개

백다은

김무연

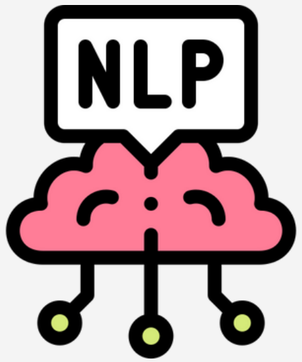


김윤주

박혜원

송여경

1. 프로젝트 배경



NLP

헬스케어

멀티모달

공통적인 관심사를 접목한 주제 선정하고자 함

음성 데이터와 텍스트 데이터를 활용한 감정 분류 모델을 구현해보자!

→ 텍스트-음성 데이터 기반 멀티모달 감정 분류

2. 데이터

2.1 사용한 데이터셋

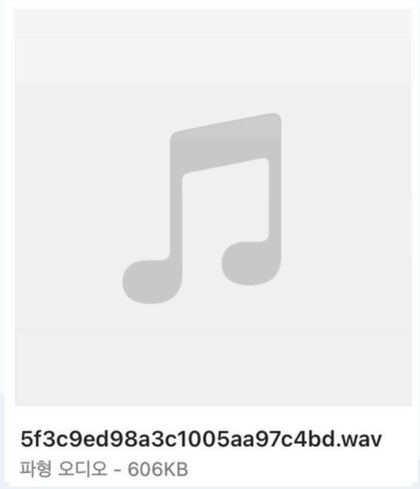
감정 분류를 위한 대화 음성 데이터셋

- **AI Hub** 제공, 다분류 감정 한국어 데이터
- 데이터셋 크기: 19,374건
- 데이터 구조: wav 파일, csv 포맷 메타 정보(발화문, 상황, 감정 및 세기)
- 데이터 정보: 감성 대화 어플리케이션을 이용해 자연스럽게 대화한 내용을 수집 후 7가지 감정(happiness, angry, disgust, fear, neutral, sadness, surprise)에 대해 사람이 직접 라벨링한 데이터

wav_id	발화문	상황	1번 감정	1번 감정세기	2번 감정	2번 감정세기	3번 감정	3번 감정세기	4번 감정	4번감정세기	5번 감정	5번 감정세기	나이	성별
5f4141e29dd513131eacee2f	헐! 나 이벤트에 당첨 됐어.	happines	angry	2	surprise	2	happines	2	happines	2	happines	2	48	female
5f4141f59dd513131eacee30	내가 좋아하는 인플루언서가 이벤트를 하더라고. 그래서 그냥 신청 한번 해봤지.	happines	neutral	0	happines	2	happines	2	happines	2	happines	2	48	female
5f4142119dd513131eacee31	한 명 뽑는 거였는데, 그게 바로 내가 된 거야.	happines	angry	2	happines	2	happines	2	happines	2	happines	2	48	female
5f4142279dd513131eacee32	당연히 마음에 드는 선물이니깐, 이벤트에 내가 신청 한번 해본 거지. 비싼 거야. 그래	happines	angry	2	happines	2	happines	2	happines	2	happines	1	48	female
5f3c9ed98a3c1005aa97c4bd	에피타이저 정말 좋아해. 그 것도 괜찮은 생각인 것 같애.	neutral	happines	2	happines	1	happines	2	happines	1	happines	1	48	female
5f3c9ef78a3c1005aa97c4be	난 부페 형식의 음식들도 정말 좋아해. 그 것도 좀 알려 줘.	neutral	neutral	0	happines	2	happines	1	happines	1	neutral	0	48	female
5f3c9f658a3c1005aa97c4c7	응. 완전히 끝난 거야. 한 달 동안 주말에 쉬지도 못하고 일만 했거든.	happines	happines	2	happines	1	sadness	1	sadness	1	sadness	1	48	female
5f3c9f808a3c1005aa97c4c8	신나는 음악 듣는 것도 좋고, 어디 여행 가고 싶고 이 것 저 것 다 해보고 싶어.	happines	neutral	0	happines	2	happines	2	happines	1	sadness	1	48	female
5f3c9f9c8a3c1005aa97c4cb	친구들도 내 연락 기다리고 있을 텐데 내가 까먹고 있었네?	happines	neutral	0	happines	1	sadness	1	sadness	1	neutral	0	48	female
5f3c9fcc8a3c1005aa97c4ce	그래. 일단은 친구들부터 만나서 여행 계획에 대해서 얘기 좀 해봐야 되겠어.	happines	neutral	0	happines	1	neutral	0	neutral	0	neutral	0	48	female
5f3ca01b8a3c1005aa97c4d3	나 요즘 너무 우울해 죽겠어.	sadness	sadness	2	sadness	2	sadness	2	sadness	1	sadness	2	48	female

텍스트 데이터

+



음성 데이터

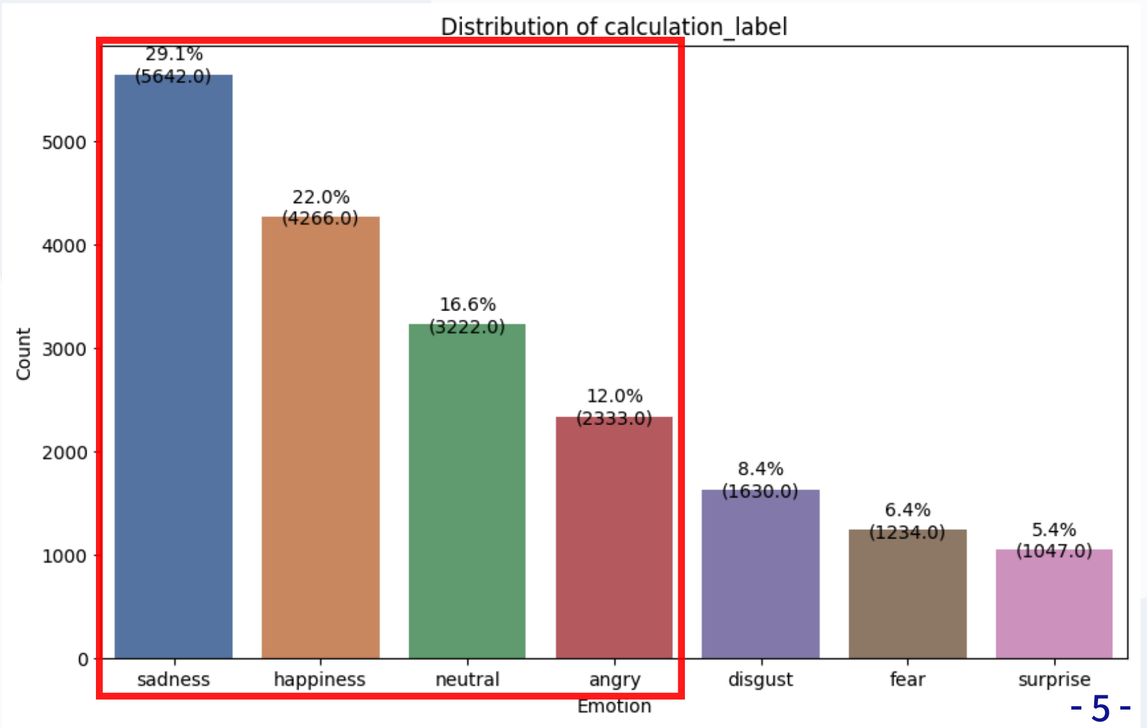
2. 데이터

2.2 데이터 전처리

감정 레이블 전처리

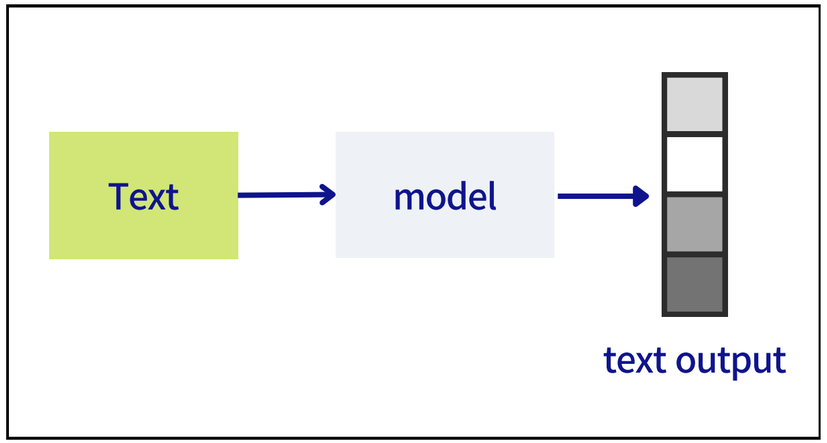
- 1번~5번 감정 label과 감정 세기를 곱한 후 더하여 가장 감정 세기가 큰 감정 label을 감정 label로 설정
- 감정 세기가 동일할 경우, 5개의 감정 중에서 가장 많이 등장한 감정에 우선순위를 부여하여 결정
- 전체 7가지 감정 중 인간이 선천적으로 느끼는 기본 정서를 가장 뚜렷하게 나타내는 4개의 감정(**sadness, happiness, neutral, angry**)만 사용

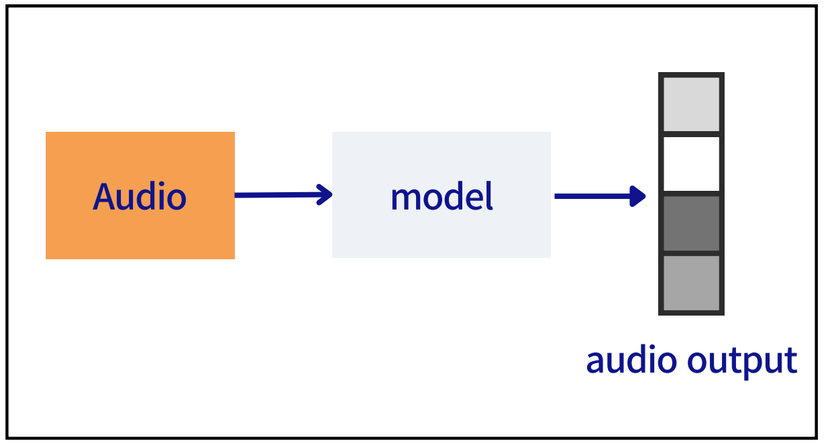
1번 감정	1번 감정세기	2번 감정	2번 감정세기	3번 감정	3번 감정세기	4번 감정	4번감정세기	5번 감정	5번 감정세기
angry	2	surprise	2	happiness	2	happiness	2	happiness	2
neutral	0	happiness	2	happiness	2	happiness	2	happiness	2
angry	2	happiness	2	happiness	2	happiness	2	happiness	2
angry	2	happiness	2	happiness	2	happiness	2	happiness	1
happiness	2	happiness	1	happiness	2	happiness	1	happiness	1
neutral	0	happiness	2	happiness	1	happiness	1	neutral	0
happiness	2	happiness	1	sadness	1	sadness	1	sadness	1
neutral	0	happiness	2	happiness	2	happiness	1	sadness	1
neutral	0	happiness	1	sadness	1	sadness	1	neutral	0
neutral	0	happiness	1	neutral	0	neutral	0	neutral	0
sadness	2	sadness	2	sadness	2	sadness	1	sadness	2
neutral	0	sadness	1	sadness	2	sadness	1	sadness	2

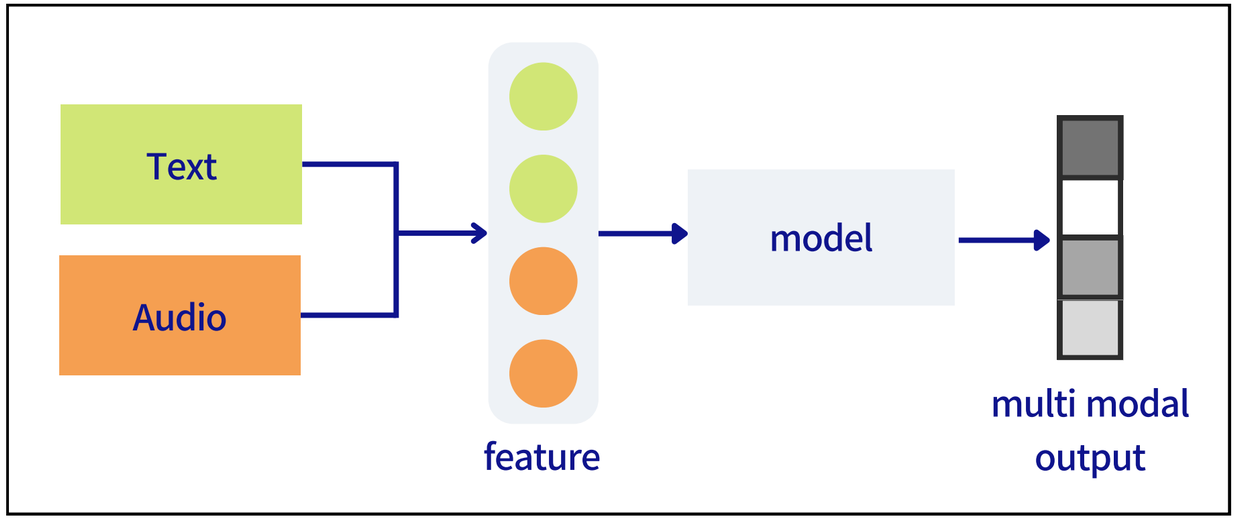


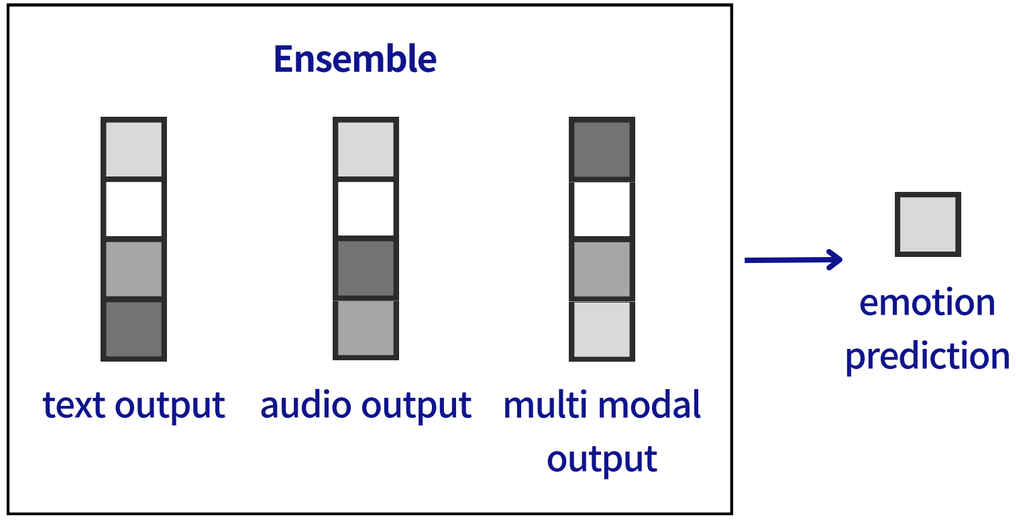
3. 모델 설명

3.1 전체 파이프라인









3. 모델 설명

3.1 멀티모달에서 활용

MFCC (Mel Frequency Cepstral Coefficients)

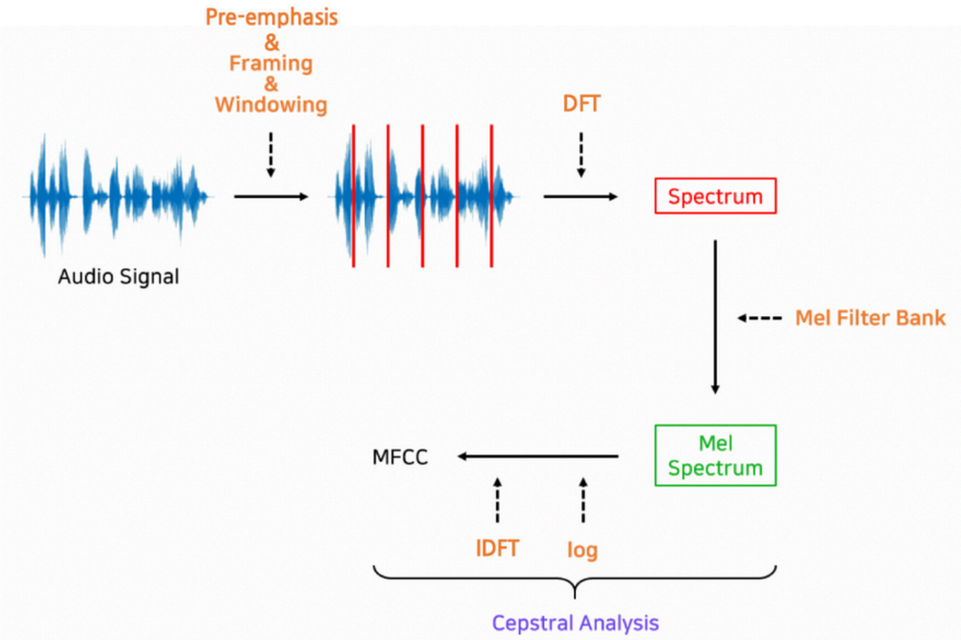
음성 및 오디오 신호 처리에서 대표적으로 사용하는 기술로, 음성데이터를 특징백터화해주는 알고리즘

① STFT(Short Time Fourier Transform)에 의해 주어진 음성 신호를 작은 프레임 단위로 나누어서 주파수 영역의 데이터로 변환

② Mel Filter Bank로 멜 스펙트럼을 계산

③ 로그 스케일링하고 DCT(Discrete Cosine Transfrom)을 수행

=> 이를 이용하여 해당 프레임의 특징을 추출



3. 모델 설명

3.2 오디오 [ResNet (Mel Spectogram)]

Mel Spectogram

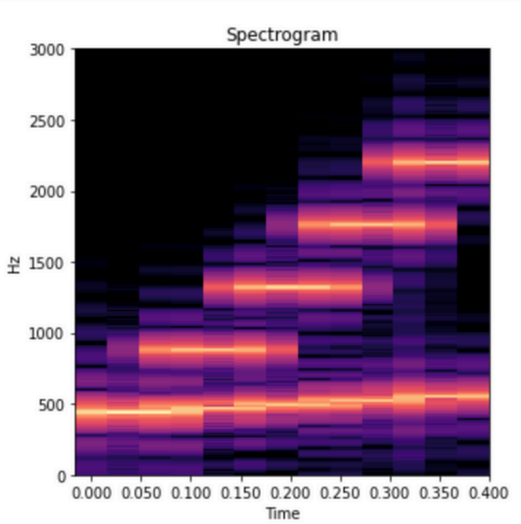
음향 신호를 분석하는 도구로, 음성이나 음악 등의 오디오 데이터를 시각적으로 표현한 것

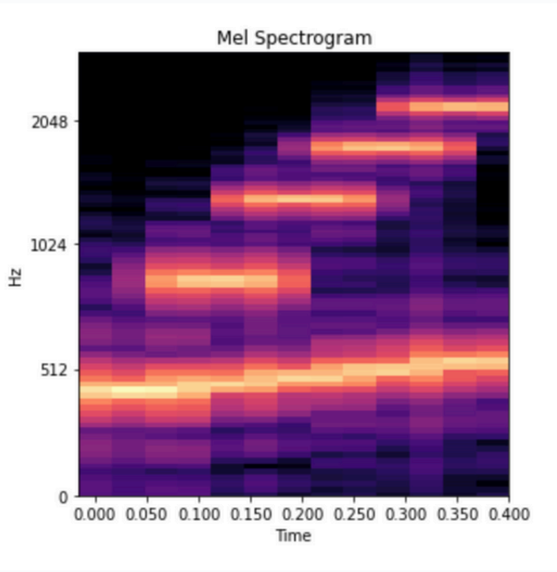
일반적인 spectogram은 시간에 따른 주파수 성분의 강도를 보여주는데,

mel spectogram은 이를 인간의 청각 특성에 맞게 변환한 것

```python
def wav_to_spectrogram(wav_file, n_fft=400, hop_length=160, n_mels=128):
    waveform, sample_rate = torchaudio.load(wav_file)
    spectrogram_transform = torchaudio.transforms.MelSpectrogram(
        sample_rate=sample_rate, n_fft=n_fft, hop_length=hop_length, n_mels=n_mels
    )
    spectrogram = spectrogram_transform(waveform)
    return spectrogram

def pad_spectrogram(spectrogram, max_len):
    c, h, w = spectrogram.size()
    if w < max_len:
        pad = max_len - w
        spectrogram = F.pad(spectrogram, (0, pad), mode='constant', value=0)
    return spectrogram
```





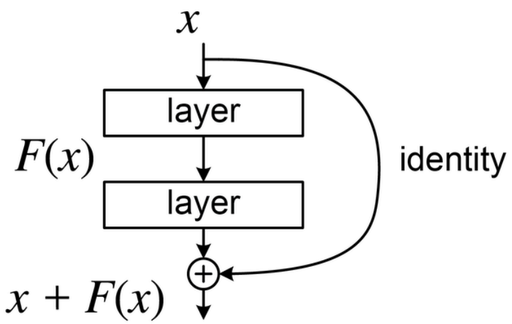
3. 모델 설명

3.2 오디오 [ResNet (MFCC)]

ResNet

① Residual Learning

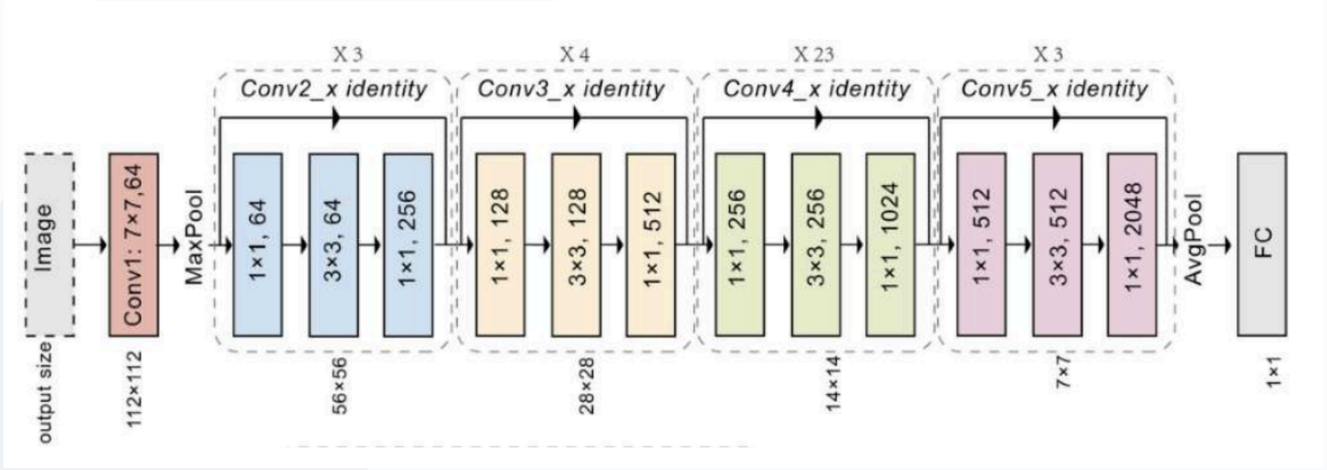
입력 layer를 다시 이용하는 residual function을 사용하여
더 쉬운 최적화와 깊은 네트워크에서의 정확도 향상이 가능



② Identity Mapping by Shortcuts

모델은 입력과 출력간의 차이를 학습하는 대신,
잔차(residual)를 학습하여 더 효과적인 학습을 이룰 수 있음

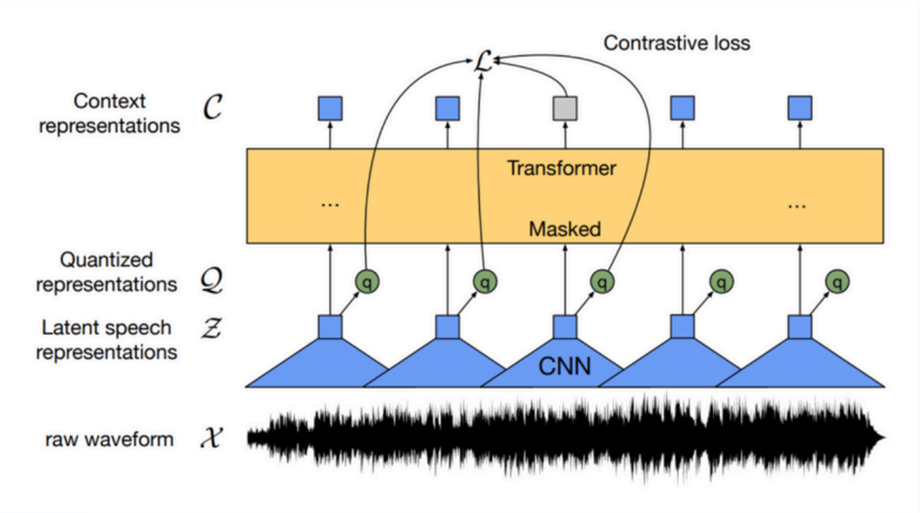
$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}.$$



3. 모델 설명

3.2 오디오 [wav2vec 2.0]

wav2vec 2.0



① Feature Encoder (X -> Z)

- multi-layer CNN으로 구성
- 원시 음성 신호 sequence 입력값인 X를 입력 받아서 매 시점마다 latent speech representation(z1,⋯,zT) 출력

② Quantization module (Z -> Q)

- 일부 Z를 quantization하여 추후 모델이 예측해야 하는 target 으로 사용
- one-hot vector로 생성

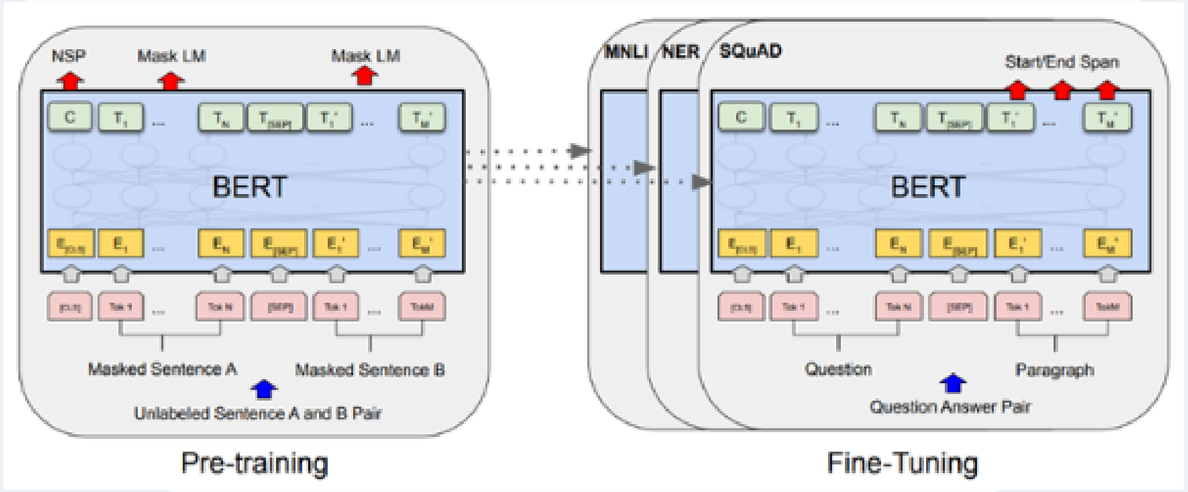
③ Contexualized Representations with Transformers (Z -> C)

- 나머지 z1,⋯,zT sequence가 입력되면, Transformer에 입력되기 전에 일부가 마스킹되어 quantization된 벡터를 예측하도록 학습
- transformer 블록에 의해 sequence 내 모든 맥락 정보가 파악된 c1,⋯,cT sequence 출력

3. 모델 설명

3.2 텍스트 - BERT의 파생모델 (KoBERT, Klue-RoBERTa, KoELECTRA)

- **BERT(Bidirectional Encoder Representation from Transformer)**
  : Transformer의 attention 기법을 이용한 embedding model



- transformer Encoder의 multi-head attention mechanism만을 사용
- 토큰 임베딩, 세그먼트 임베딩, 위치 임베딩을 합산하여 BERT의 입력으로 제공됨
- 거대한 말뭉치를 기반으로 MLM, NSP task를 동시에 사용하여 사전 학습
- MLM : 주어진 입력 문장 중 전체 단어의 15%를 무작위로 마스킹하고 이를 예측하도록 모델을 학습
- NSP : 두 문장을 입력하고 두 번째 문장이 첫 번째 문장의 다음 문장인지 예측하도록 하는 이진 분류 task

3. 모델 설명

3.2 텍스트 (KoBERT)

- 구글 BERT base multilingual cased의 한국어 성능 한계로 SKT-Brain에서 한국어 자연어 처리를 위해 최적화한 BERT 기반 모델
- training set - data : 한국어 위키 / 문장 : 5M / 단어 : 54M
- batchsize : 16 / lr : 2e-5 로 text model hyperparameters 통일

```
!pip install git+https://git@github.com/SKTBrain/KoBERT.git@master
!pip install 'git+https://github.com/SKTBrain/KoBERT.git#egg=kobert_tokenizer&subdirectory=kobert_hf'
```

```
# KoBERT tokenizer 및 모델 불러오기
from kobert_tokenizer import KoBERTTokenizer
from transformers import BertModel
from transformers import AdamW
from transformers.optimization import get_cosine_schedule_with_warmup
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
tokenizer = KoBERTTokenizer.from_pretrained('skt/kobert-base-v1')
bertmodel = BertModel.from_pretrained('skt/kobert-base-v1', return_dict=False)
vocab = nlp.vocab.BERTVocab.from_sentencepiece(tokenizer.vocab_file, padding_token='[PAD]')
```

3. 모델 설명

3.2 텍스트 (RoBERTa)

RoBERTa : Robustly Optimized BERT pre-training Approach

- MLM task에서 dymamic masking 적용
- BERT와 달리 NSP task 제거
- 배치 크기를 키워 학습 속도 및 모델 성능 향상
- tokenizer : larger byte-level BPE
- original BERT보다 훨씬 큰 데이터셋 사용 16GB -> 160GB

Klue-Roberta : KLUE(Korean Language Understanding Evaluation) 한국어에 특화된 대규모 텍스트 데이터를 사용하여 학습됨.

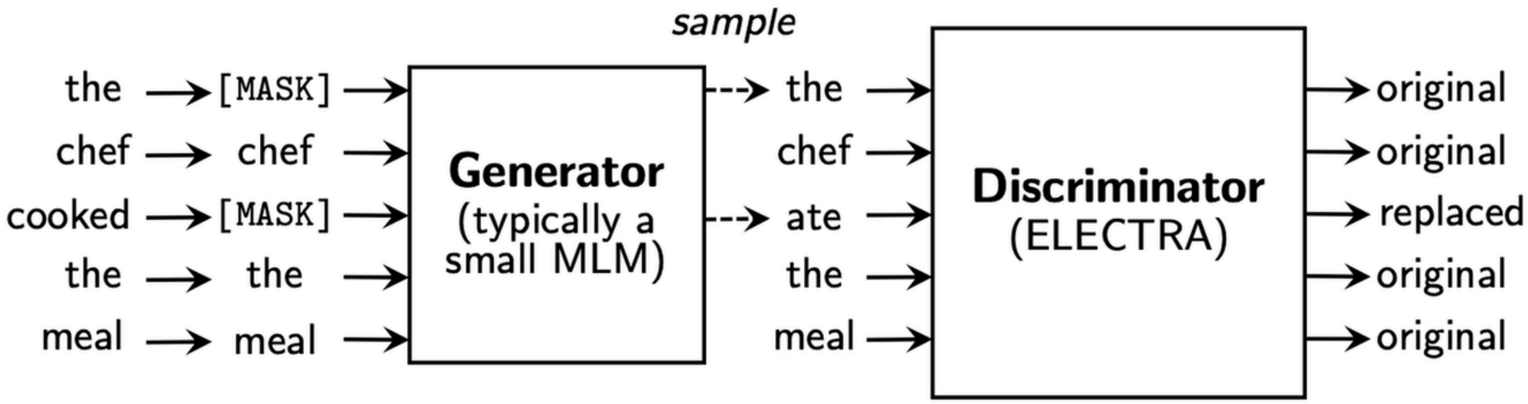
```
from transformers import AutoModel, AutoTokenizer


model = AutoModel.from_pretrained("klue/roberta-base")
tokenizer = AutoTokenizer.from_pretrained("klue/roberta-base")
```

3. 모델 설명

3.2 텍스트 (KoELECTRA)

Electra : Efficiently Learning an Encoder that Classifies Token Replacements Accurately



기존의 BERT와 같은 MLM 학습이 아닌Generator(G), discriminator(D)를 이용한 RTD(Replaced Token Detection) pre-training 학습

실제 orgin 데이터(input token)을 Generator 에서 replaced token, 혹은 origin token으로 바꾸고,
이를 Discriminator 에서 이게 origin token 인지 replaced token 인지에 대해 판별하는 프로세스

3. 모델 설명

3.2 텍스트 (KoELECTRA)

KoELECTRA - ELECTRA의 한국어버전 language model

Pretraining Details

Model	Batch Size	Train Steps	LR	Max Seq Len	Generator Size	Train Time
Base v1,2	256	700K	2e-4	512	0.33	7d
Base v3	256	1.5M	2e-4	512	0.33	14d
Small v1,2	512	300K	5e-4	512	1.0	3d
Small v3	512	800K	5e-4	512	1.0	7d

Model : KoELECTRA-base / KoELECTRA-Small

Version : v1,2,3

사용 모델 : KoELECTRA-base v3

```
koelectra_tokenizer = ElectraTokenizer.from_pretrained("monologg/koelectra-base-discriminator")
koelectra_model = ElectraForSequenceClassification.from_pretrained("monologg/koelectra-base-discriminator", num_labels=4).to(device)
```

3. 모델 설명

3.3 멀티모달 (RoBERTa, KoELECTRA)

1st Step.음성 데이터 처리

- **data augmentation**

  noise : random noise 추가
  stretch: 시간적으로 음성 데이터를 늘리기, 축소하기
  pitch: 음성의 피치 변경

```
def noise(data):
    noise_amp = 0.005 * np.random.uniform() * np.amax(data)
    return data + noise_amp * np.random.normal(size=data.shape)

def stretch(data, rate=0.8):
    return librosa.effects.time_stretch(y=data, rate=rate)

def pitch(data, sampling_rate, pitch_factor=0.7):
    return librosa.effects.pitch_shift(y=data, sr=sampling_rate, n_steps=pitch_factor)

def extract_features(data, sample_rate):
    # Using Mel spectrogram instead of MFCC
    mel_spectrogram = librosa.feature.melspectrogram(y=data, sr=sample_rate, n_mels=13)
    mel_spectrogram_db = librosa.power_to_db(mel_spectrogram, ref=np.max)
    mel_mean = mel_spectrogram_db.mean(axis=1)
    return mel_mean
```

- **extract_features**

  음성 데이터의
  Mel_spectogram 추출
  librosa.feature.melspectrogram를 사용해 오디오의 Mel_spectogram계산

2nd Step. 특성 추출 후, 데이터 프레임 생성

문장과 레이블 추가해서 하나의 벡터로 결합하여 최종 df 생성

3. 모델 설명

3.3 멀티모달 (RoBERTa, KoELECTRA)

3rd Step. train, test 데이터셋 분리

- scaler 사용(표준화)

4th Step.custom_model 분석

1D 합성곱 신경망(CNN) 기반의 감정 분류 모델

Conv1D(256, kernel_size=5, strides=1) 256개의 필터 사용

padding = ‘same’ 입력데이터와 출력 데이터 크기 동일하게 유지

activation = ‘relu’ 비선형성

Dropout(0.2) 뉴런 무작위 비활성화 - 과적합 방지

Dense(units=32, activation='relu') 출력 Layer: Softmax

5th Step. 토크나이저 로딩/ 텍스트 임베딩

- klue/roberta-base: AutoModel, AutoTokenizer를 사용.
- monologg/koelectra-base-discriminator: ElectraForPreTraining과 ElectraTokenizer 사용

3. 모델 설명

3.3 멀티모달 (RoBERTa, KoELECTRA)

6th Step.학습

- scaler 사용(표준화)

- 임베딩 생성: 각 사전 학습된 모델에 대해 텍스트 임베딩을 생성
- 데이터 분리: train_test_split훈련 세트, 테스트 세트로 분리
- 데이터 표준화: StandardScaler를 사용해 훈련 및 테스트 데이터를 표준화
- 차원 확장: 입력 데이터를 모델이 처리할 수 있는 형태로 차원을 확장
- 모델 훈련: custom_model을 사용해 모델을 빌드하고, 훈련을 진행
- 모델 평가: 테스트 데이터를 사용해 모델의 성능을 평가하고 정확도를 출력

batch_size=64, epochs=10로 통일 후 진행

```
Pre-trained Model:  klue/roberta-base
Test Accuracy:  0.5437144041061401
```

```
Pre-trained Model:  monologg/koelectra-base-discriminator
Test Accuracy:  0.5046559572219849
```

4. 실험 결과

- 텍스트 단일 모델

	KoELECTRA	KLUE-RoBerta
accuracy	0.68	**0.70**
F1-score	0.66	**0.69**

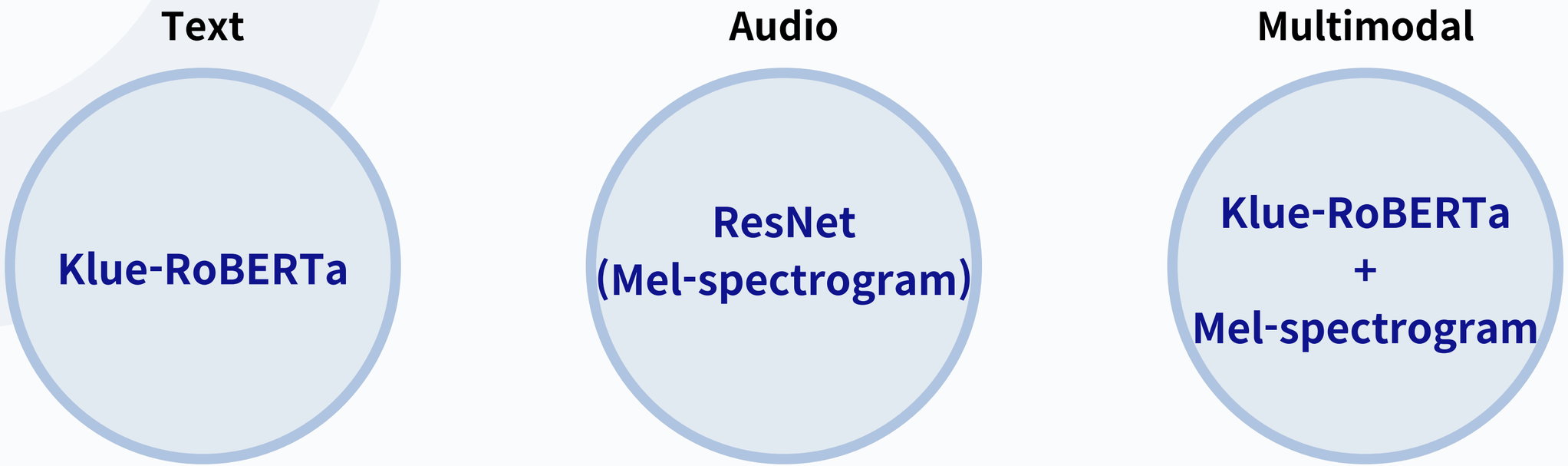
- 오디오 단일 모델

	ResNet_Mel-spectrogram
accuracy	**0.3679**
F1-score	**0.2437**

- 멀티모달

	KoELECTRA_Mel	KLUE-RoBerta_Mel
accuracy	0.36792758107185364	**0.6117038726806641**
F1-score	0.1979208623126626	**0.5977136346400203**

4.1 Ensemble

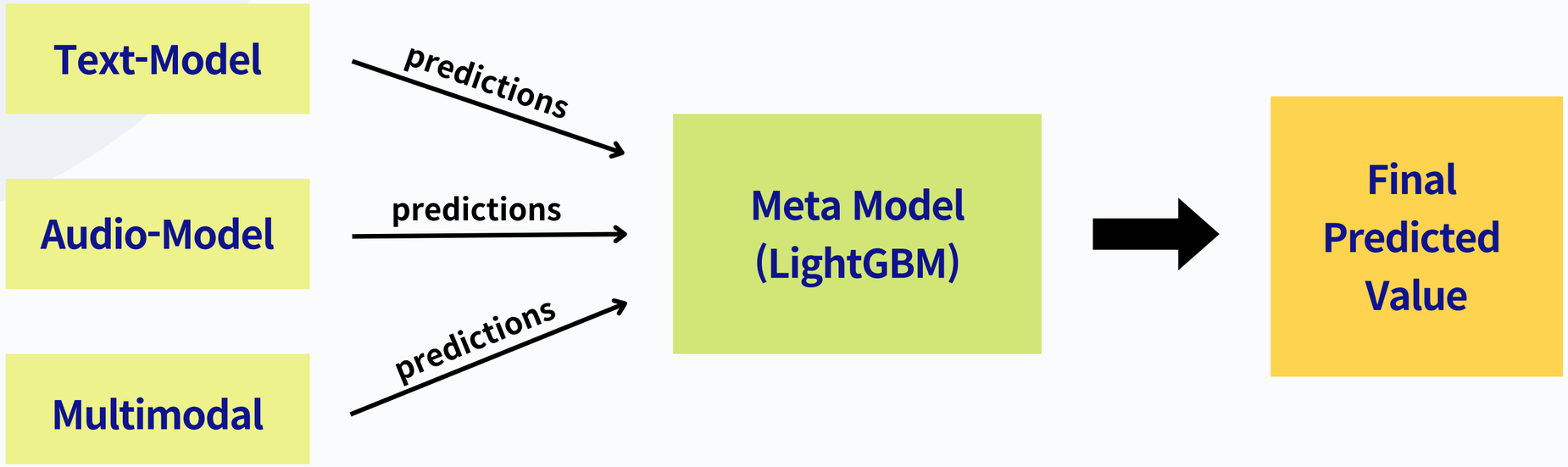


성능 향상을 목표로 각 모듈 별 가장 높은 성능을 보이는 모델을 채택하여 Ensemble 수행

4.1 Ensemble

- **Stacking**

여러 가지 모델들의 예측값을 결합하여 최종 모델의 학습 데이터로 사용해 예측하는 방법



- Accuracy : 0.6133
- F1_score : 0.5978

5. 의의 및 한계

의의

- 감정 데이터(음성, 텍스트)를 이용해서 멀티모달에 대해 학습하고 구현
- 텍스트 및 음성 처리 모델에 대해 학습할 수 있었음
- 앙상블을 통해 성능 개선

한계

- 리소스 문제로 인해 초기에 설정한 모델을 모두 돌려보지 못함
- 기대한 것보다 좋지 않은 성능을 보여 코드 개선이 필요해보임

감사합니다

Thank you