

# 2023

## 군장병 부문

# 국방 AI 경진대회

# 결과물 발표평가



팀명 | **별공지능**



대한민국 국방부  
Ministry of National Defense



과학기술정보통신부  
Ministry of Science and ICT

# I 팀 소개

팀장



대위 오 세 흥

육군사관학교  
AI · 데이터과학과



4학년 정 우 진

육군사관학교  
컴퓨터과학과



4학년 김 지 훈

육군사관학교  
AI · 데이터과학과



2학년 김 채 민

육군사관학교  
컴퓨터과학과

# II 발표 내용\_ [평가 항목] 문제 이해도

## 문 제

### 음성인식 기술을 활용한 효과적인 지휘관의 전술 명령 전달

전장에서 지휘관의 명령 및 의도 전달은 매우 중요

-> 군사 작전 중 소음 및 통신불량 으로 인하여 음성 명령 전달 안되는 경우 다

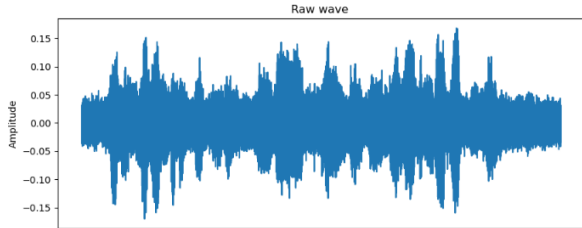
-> 문제 해결 위해 음성을 전사하는 모델 필요성 대두

## 해결 과제1

### 음성 파일의 전처리

제공된 테스트 파일에 노이즈가 켜 있음

->오디오 파일을 Denoise 하는 전처리 과정 필요



[1] 노이즈가 포함된 audio\_0.wav 파일의 그래프

## 해결 과제2

### 음성 파일을 텍스트 파일로 변환

음성 파일을 텍스트 파일로 변환하는 STT(Speech To Text) 모델 구현 필요

->Whisper을 이용하여 STT 모델 구현

#### 음성 인식



사람의 음성으로부터 발화한 텍스트를 읽어내는 기술

[2] 음성을 텍스트로 변환하는 stt 기술

출처 : <https://velog.io/@kyungmin1029/NLP-%ED%95%9C%EA%B5%AD%EC%96%B4-%EC%9D%8C%EC%84%B1-%EC%9D%B8%EC%8B%9D-KoSpeech>

## 해결 과제3

### 결과값 정확도 향상

정확히 인식되지 않는 부분에 대한 결과값 정확도 향상 필요  
모델 학습 시 혼동을 줄 수 있는 학습 값 제거

```
# 예시 DataFrame 생성
from collections import Counter
# 'text' 컬럼에서 한글 문자 개수 및 종류 확인
total_characters = submission['text'].str.replace('[^가-힣]', '', regex=True).str.len().sum()
unique_characters = set(''.join(submission[:945]['text']).replace(' ', ''))

# print(각 줄 문자 개수: {total_characters})
print(각 줄 종류: {unique_characters})
```

[3] 한국어 이외의 문자(일본어, 한자, 특수기호 출력 확인

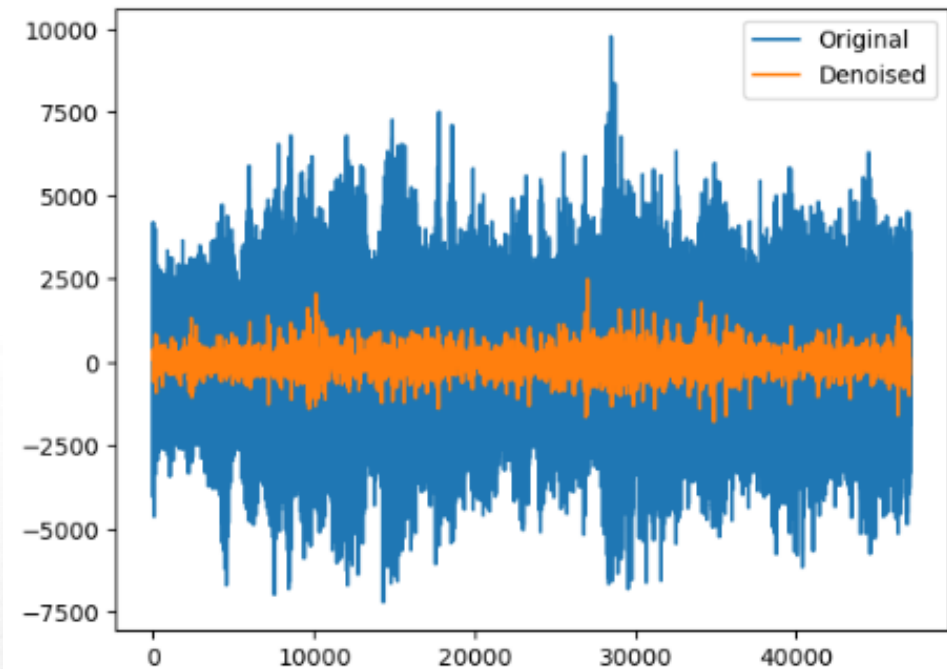
# II 발표 내용\_ [평가 항목] 데이터 활용

## 음성 파일 전처리 후 학습 활용

음성 내의 노이즈로 인해 음성이 정확하게 인식되지 않는 문제 해결

-> Denoise 코드 구현 (초기 코드)

-> scipy.io와 Noisereduce 라이브러리 사용



[4] Denoise

## 음성 데이터, 제공된 텍스트로 모델 학습

-> Whisper 모델 사용

```
from transformers import Seq2SeqTrainer, TrainerCallback, TrainingArguments, TrainerState, TrainerControl
from transformers.trainer_utils import PREFIX_CHECKPOINT_DIR

# This callback helps to save only the adapter weights and remove the base model weights.
class SavePeftModelCallback(TrainerCallback):
    def on_save(
        self,
        args: TrainingArguments,
        state: TrainerState,
        control: TrainerControl,
        **kwargs,
    ):
        checkpoint_folder = os.path.join(args.output_dir, f"{PREFIX_CHECKPOINT_DIR}-{state.global_step}")

        peft_model_path = os.path.join(checkpoint_folder, "adapter_model")
        kwargs["model"].save_pretrained(peft_model_path)

        pytorch_model_path = os.path.join(checkpoint_folder, "pytorch_model.bin")
        if os.path.exists(pytorch_model_path):
            os.remove(pytorch_model_path)
        return control

trainer = Seq2SeqTrainer(
    args=training_args,
    model=model,
    train_dataset=common_voice["train"],
    eval_dataset=common_voice["test"],
    data_collator=data_collator,
    compute_metrics=compute_metrics,
    tokenizer=processor.feature_extractor,
    callbacks=[SavePeftModelCallback],
)

model.config.use_cache = False # silence the warnings. Please re-enable for inference!
```

[5] Hugging Face의 Transformers 라이브러리를 사용 Seq2Seq 모델 학습

# II 발표 내용\_ [평가 항목] 모델 완성도

## 모델 완성도 향상 방법1

### 데이터 Augmentation

두가지의 디노이징 방법으로 트레이닝 데이터 증가

### 훈련 데이터 선별

긴 데이터 제거, 특수문자 제거, 한글화

## 모델 완성도 향상 방법2

### 사용가능한 Whisper 버전

Whisper Large, Medium, tiny 등 모델 사이즈 조정  
-> 모델 사이즈에 따라 해석 달라짐

Whisper ver-2, ver-3 바꿈에 따라 결과값 달라짐

-> 최적의 모델 찾기

## 모델 평가

멀공지능



0.764609

2023. 12. 1. 08:47



# II 발표 내용\_ [평가 항목] 수행 프로세스

## 1 오디오 전처리

```
import noisereducer
from scipy.io import wavfile
import noisereducer as nr
from tqdm import tqdm

#load data
for i in tqdm(range(1000)):
    rate, data = wavfile.read(f"./dataset_file/train/audio_{i}.wav")
    #perform noise reduction
    reduced_noise=nr.reduce_noise(y=data, sr=rate)
    wavfile.write(f"./denoised_train/audio_{i}.wav", rate,reduced_noise)
```

-> [13] 오디오 denoise 방법 고안

- 1. Wiener filter**  
오히려 노이즈 증가하는 경향
- 2. FFT**  
음성 명확하게 해석 안됨
- 3. Noisereducer 모듈**  
노이즈 거의 안 들리는 대신 소리 작음
- 4. Butterworth filter**  
가장 노이즈가 잘 제거

## 2 모델 학습

### 가장 최적화 된 모델 나올 때까지 시도

디노이즈 기법 바뀌가며 테스트 세트 변화  
-> 모델에 넣어가며 학습

전처리(필요 없는 문자 제거,  
오디오 노이즈 최소화)

Whisper Large, Medium,  
버전 바뀌가며 조정

## 3 정확도 향상

### 1. 테스트 음성 파일 변화

```
from scipy import signal
import random

def f_high(y,sr):
    print(sr)
    #b,a = signal.butter(10, 2000/(sr/2), btype='highpass')
    #b, a = signal.butter(10, [500/(sr/2),1000/(sr/2)], btype='band')
    b,a = signal.butter(2, 0.1, output='ba')
    yf = signal.lfilter(b,a,y)
    return yf
```

[14] Butterworth filter 사용해서 디노이즈한 테스트 파일 생성

### 2. 불필요 문자 제거, 정규화

### 3. Whisper 사이즈, 버전 조정

```
from transformers import WhisperFeatureExtractor

feature_extractor = WhisperFeatureExtractor.from_pretrained("distil-whisper/distil-medium.en")

from transformers import WhisperTokenizer

tokenizer = WhisperTokenizer.from_pretrained("distil-whisper/distil-medium.en", language="Korean", task="transcribe")
```

[16] Whisper 사이즈 Medium으로 설정한 경우



# 2023 국방 AI 경진대회



## Where AI & Defense Connect

### 소감 한마디

오세홍

지금까지 경진대회라는 것에 참여를 해본적이 없었는데, 국방AI경진대회에 참여하면서 군의 인재들과 함께 할 수 있어서 뜻 깊은 시간이었습니다.

정우진

우물 안 개구리가 되지 말아야겠다는 생각이 들었다. 앞으로의 학업에 큰 동기부여가 되었다.

수명이 2년 3개월은 줄은 것 같지만 올해 들어 가장 몰두한 2박 3일의 경험

김지훈

AI 경진대회에 처음 참가하고 음성 데이터도 처음 다뤄봐서 신기했다. 더 많이 공부해야겠다고 느꼈다.

김채민