잡음제거 & 문자 표현을 통한 정확한 의사 전달에 관한 연구(STT)

by 연수생 김현수(2024.12)

목차

1. 개요
2. 목적
3. 연구
4. 결론

잡음제거 & 문자 표현으로 정확한 의사 전달에 관한 연구

- 음성 인식 기술의 발전에도 불구하고, 실제 환경에서의 정확한 의사 전달은 여전히 도전 과제로 남아있다.

- 본 연구에서는 발화자의 음성에서 잡음을 제거하고 이를 정확한 문자로 표현하여 의사 전달의 정확성을 높이는 혁신적인 접근 방식을 제안한다.

- 연구 접근 방식은 온라인상에 배포되어 있는 인공지능 모델을 활용한다. ECAPA를 통해 화자 구분, Whisper를 통해 언어 감지 및 “음성 to 텍스트” 변환, 그리고 TSCN 모델을 통한 잡음 제거 기술을 결합하여 무전기 환경에서의 의사소통 문제를 해결하고자 한다.

왜 필요하며 어디에 적용할 것인가?

군대에서 아군 식별과 정확한 의사소통은 매우 중요하다.
특히 다음과 같은 상황에서 기술의 필요성이 두드러진다.

1 무전기 환경

백색잡음 상에서 중요한 대화를 나눌 때 화자 인식을 하고 소음으로 인한 의사소통 장애를 방지하는 데 도움이 된다.

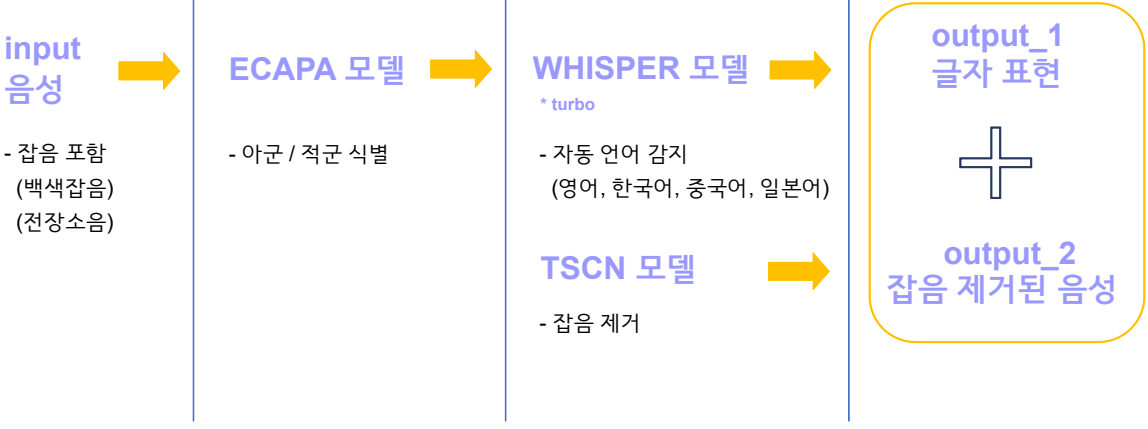
2 화상 회의

화상 회의나 원격 교육 시 음질 저하로 인한 의사소통 문제를 해결할 수 있다.

3 유선 전화

선로 노후화로 인한 의사소통 문제 해결하는데 도움이 된다.

알고리즘 및 시스템 개요


input
음성
- 잡음 포함
(백색잡음)
(전장소음)
ECAPA 모델
- 아군 / 적군 식별
WHISPER 모델
* turbo
- 자동 언어 감지
(영어, 한국어, 중국어, 일본어)
TSCN 모델
- 잡음 제거
output_1
글자 표현
output_2
잡음 제거된 음성

알고리즘: ECAPA 소개_1

VS

- 안녕하세요

- 반갑습니다.

같은 사람 다른 발화

같은 사람이 다른 말을 하여도 같은 사람인 지 구별한다.

VS

- 안녕하세요

- 안녕하세요

다른 사람 같은 발화

서로 다른 사람이 같은 말을 하여도 같은 사람이 아닌 것을 구별한다.

알고리즘: ECAPA 소개_2

ECAPA(Emphasized Channel Attention, Propagation and Aggregation)

Table 1: *EER and MinDCF performance of all systems on the standard VoxCeleb1 and VoxSRC 2019 test sets.*

Architecture	# Params	VoxCeleb1		VoxCeleb1-E		VoxCeleb1-H		VoxSRC19
		EER(%)	MinDCF	EER(%)	MinDCF	EER(%)	MinDCF	EER(%)
E-TDNN	6.8M	1.49	0.1604	1.61	0.1712	2.69	0.2419	1.81
E-TDNN (large)	20.4M	1.26	0.1399	1.37	0.1487	2.35	0.2153	1.61
ResNet18	13.8M	1.47	0.1772	1.60	0.1789	2.88	0.2672	1.97
ResNet34	23.9M	1.19	0.1592	1.33	0.1560	2.46	0.2288	1.57
ECAPA-TDNN (C=512)	6.2M	1.01	0.1274	1.24	0.1418	2.32	0.2181	1.32
ECAPA-TDNN (C=1024)	14.7M	**0.87**	**0.1066**	**1.12**	**0.1318**	**2.12**	**0.2101**	**1.22**

출처: https://arxiv.org/pdf/2005.0714
3

1. 화자 구분에 탁월한 성능을 보인다.
2. 약간의 잡음이 포함되어 있어도 EER 10% 미만의 성능을 보인다.

알고리즘: TSCN 소개

TSCN

딥러닝 기반 소음 제거 모델(Deep Noise Suppression)로 ICASSP 2021 DNS Challenge의 트랙1(실시간 소음감소)에서 1위를 차지한 모델.

입력 https://user-images.githubusercontent.com/65753560/143393711-c9ec37a0-95ef-407f-8e72-444553c43bc0.mp4

출력 https://user-images.githubusercontent.com/65753560/143393778-9dc9331c-915a-4555-b4f8-4197a575420f.mp4

정답 https://user-images.githubusercontent.com/65753560/143393794-f40d689c-9892-49bc-81d4-c28a3a5aeb18.mp4

알고리즘: whisper 소개

openai/whisper-large-v3-turbo

OpenAI에서 제작한 자동 음성 인식 및 문자표현을 수행하는 최신 모델.
관련링크: https://huggingface.co/openai/whisper-large-v3-turbo

Size	Parameters	English-only	Multilingual
tiny	39 M	✓	✓
base	74 M	✓	✓
small	244 M	✓	✓
medium	769 M	✓	✓
large	1550 M	x	✓
large-v2	1550 M	x	✓
large-v3	1550 M	x	✓
large-v3-turbo	809 M	x	✓

실험[ECAPA]

다양한 조건에서 시스템의 성능을 평가.

실험	조건	결과(EER)
화자 구분	백색잡음 - 0.025	4.89%
화자 구분	백색잡음 - 0.05	10.93%
화자 구분	백색잡음 - 0.1	22.20%
화자 구분	생활소음 - 0.25	4.52%
화자 구분	생활소음 - 0.5	9.88%
화자 구분	생활소음 - 1	18.87%

흔히 접할 수 있는 잡음 크기를 선정하여 EER 결과로 확인.

잡음이 있음에도 불구하고 화자 구분을 잘 하는 것으로 판단.

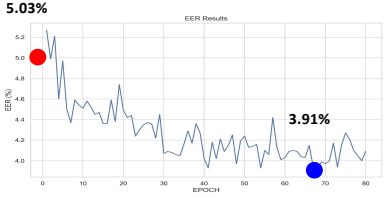
* EER 수치가 낮을수록 성능이 우수함.

* 백색잡음 - 숫자 : 잡음의 크기를 조절하는 파라미터값. 0.025일 때 일반적인 무전기에서 나오는 잡음과 비슷함.

* 전장소음 - 숫자 : 잡음의 크기를 조절하는 파라미터값. 1일 때 목소리 dB과 동일. 0.5일 때 목소리 dB의 1/2 크기

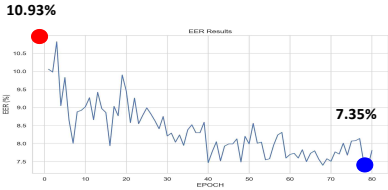
TRAIN

백색잡음 0.025



잡음 세기	0.025	0.05	0.1	0.2
사전 학습 모델	5.03 %(80)	10.93 %(80)	22.20 %(80)	36.48 %(80)
Train 도전 1	3.91 %(68)	**7.35 %(79)**	**15.52 %(40)**	**29.54 %(40)**
Train 도전 2	4.00 %(72)	8.05 %(53)	17.65 %(47)	31.74 %(56)
Train 도전 3	**3.83 %(78)**	7.50 %(80)	15.78 %(53)	29.99 %(46)

백색잡음 0.05



- 필요에 의해 잡음을 학습을 할 수도 있다.
- Pretrain model을 활용하여 사용해도 좋지만, 잡음에 대해 학습을 한 뒤에 활용하면 더욱 높은 정확도로 목소리를 구별할 수 있다.

실험[whisper]

```python
import librosa
from transformers import WhisperProcessor, WhisperForConditionalGeneration
from datasets import load_dataset

# load model and processor
processor = WhisperProcessor.from_pretrained("openai/whisper-large-v3-turbo")
model = WhisperForConditionalGeneration.from_pretrained("openai/whisper-large-v3-turbo")

# 1은 원본 2는 잡음. 테스트용
audio_path_1 = "00001_ori.wav"
audio_array_1, sampling_rate_1 = librosa.load(audio_path_1, sr=16000)
input_features_1 = processor(audio_array_1, sampling_rate=sampling_rate_1, return_tensors="pt").input_features

audio_path_2 = "00001_n_0.025.wav"
audio_array_2, sampling_rate_2 = librosa.load(audio_path_2, sr=16000)
input_features_2 = processor(audio_array_2, sampling_rate=sampling_rate_2, return_tensors="pt").input_features

audio_path_3 = "00001_dn_0.025.wav"
audio_array_3, sampling_rate_3 = librosa.load(audio_path_3, sr=16000)
input_features_3 = processor(audio_array_3, sampling_rate=sampling_rate_3, return_tensors="pt").input_features

predicted_ids_1 = model.generate(input_features_1)
predicted_ids_2 = model.generate(input_features_2)
predicted_ids_3 = model.generate(input_features_3)

transcription_1 = processor.batch_decode(predicted_ids_1, skip_special_tokens=True)
transcription_2 = processor.batch_decode(predicted_ids_2, skip_special_tokens=True)
transcription_3 = processor.batch_decode(predicted_ids_3, skip_special_tokens=True)

print(transcription_1)
print(transcription_2)
print(transcription_3)
```

```
[" Very often I am, and then sometimes I'm not. And when I catch myself realizing that I have reverted back into being Eartha May,"]
[" Very often I am, and then sometimes I'm not. And when I catch myself realizing that I have reverted back into being Earth's May,"]
[" Very often I am and then sometimes I'm not. And when I catch myself realizing that I have reverted back into being Earth and A,"]
```

whisper-large-v3-turbo 모델을 가져와 음성을 텍스트로 표현하는 코드

약간의 잡음이 들어가 있는 상태에서도 텍스트로 잘 표현하는 것을 발견.

1번줄 : 원본 음성을 글자로 표현
2번줄 : 잡음이 들어간 음성을 글자로 표현
3번줄 : TSCN으로 잡음을 제거하고 글자로 표현

실험[TSCN]

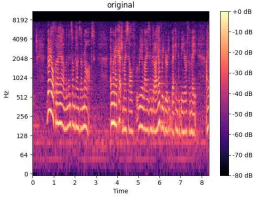
실험	조건	결과(EER)	TSCN 적용	결과(EER)
화자 구분	백색잡음 - 0.025	5.03%	x	x
화자 구분	백색잡음 - 0.05	10.93%	→	11.91%
화자 구분	백색잡음 - 0.1	22.20%	→	19.99%
화자 구분	생활소음 - 0.25	4.52%	x	x
화자 구분	생활소음 - 0.5	9.88%	x	x
화자 구분	생활소음 - 1	18.87%	x	x

성능 하락

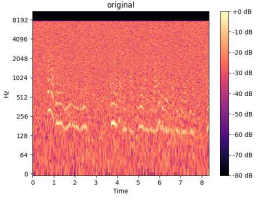
성능 향상.
But, 유의미 X

TSCN 모델을 활용하여 잡음을 제거한 후 ECAPA 모델로 화자 구분을 시도했을 때 EER 값은 오히려 증가하여 성능이 하락하는 모습을 보임.
즉, 잡음을 제거하면 원본 음성의 패턴과 텐서값에 영향을 주어 1 VS 1 비교에서 올바르게 작동하지 않다는 결론을 얻음.
따라서, ECAPA로 화자 구분을 할 때, TSCN을 활용한 뒤 구분하는 것은 적절치 않다.

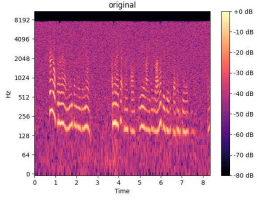
실험[TSCN_2]



원본음성



원본음성 + 백색잡음



원본음성 + 백색잡음
TSCN 잡음제거

TSCN 모델로 잡음을 제거하면 ECAPA 성능 향상에는 **좋지 않지만**,
위의 멜스펙토그램으로 보이듯이 실제로 사람이 소리를 청취하여 비교하면 개선된 소리가 들림.
따라서, ECAPA로 화자 구분 뒤에 전화기상의 수신자가 들을 때는 잡음을 제거한 뒤 청취할 수 있도록 시스템 개발이 필요.

필요 개선사항

지금까지의 연구결과는 매우 유용하지만, 시스템의 실용화와 성능 향상을 위해 다음과 같은 사항들이 필요하다.

대규모 데이터셋

다양한 언어, 방언, 잡음 환경을 포함한 군 전용 대규모 음성 데이터셋의 구축이 필요

계산 자원

실시간 처리를 위한 고성능 GPU 하드웨어가 필요

알고리즘 개선

연구 환경보다 더 극단적인 잡음 환경에서의 성능 향상을 위한 추가적인 알고리즘 연구가 필요.
(기존 모델에 추가 데이터셋 학습)

이동기기 최적화

이동기기에서의 효율적인 실행을 위한 모델 경량화 및 최적화가 필요