

Zero-Speech Challenge

지도교수 : 조성배 교수님

지도조교 : 문형준 조교님

Team : Deep Sound

위준복 신명진 조형진

목차

1. 연구 주제
2. 기존 연구의 한계점
3. 제안하는 연구의 소개
4. 개선 방법론
5. 연구 방법
6. 연구 결과
7. 팀의 구성 및 역할
8. 참고 문헌

1. 연구 주제 : Zero-Speech Challenge - Speech Feature 기반 waveform 생성에 대한 연구

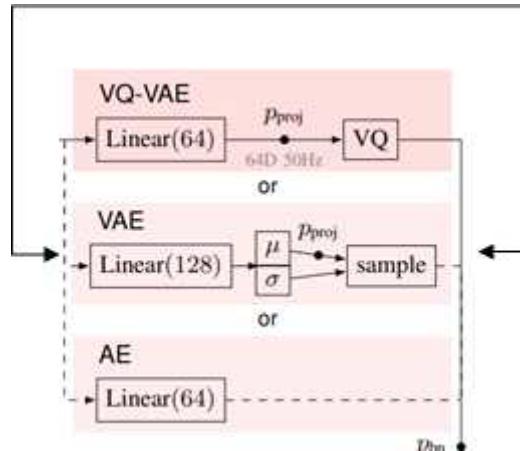
- 사람이 말하는 음성을 받아서 말하는 내용을 파악한 뒤(Speech Feature) 다른 사람이 말하는 것처럼 음성 발화 데이터를 합성(waveform 생성)하는 것을 목표로 하는 딥러닝 기술에 대한 연구입니다.

- 전통적 언어 기술은 방대한 양의 텍스트 및 언어 지식을 기반으로 훈련됩니다. 이는 텍스트 데이터나 전문가 리소스가 없는 언어에는 적용할 수 없는 방식이기 때문에 자료가 부족한 언어 문서화 및 서비스를 제공에 있어서 연구 가치가 있습니다.

- 영어가 원시 감각을 통해 자발적으로 언어 습득하듯이 딥러닝 모델 또한 비슷하게 비감독 학습하도록 구현하는 것이 목표입니다. 이를 통해 언어 발달 예측 모델을 기대할 수 있습니다.

2. 기존 연구와 한계점

a. 기존 연구 : VQ-VAE 모델과 배경

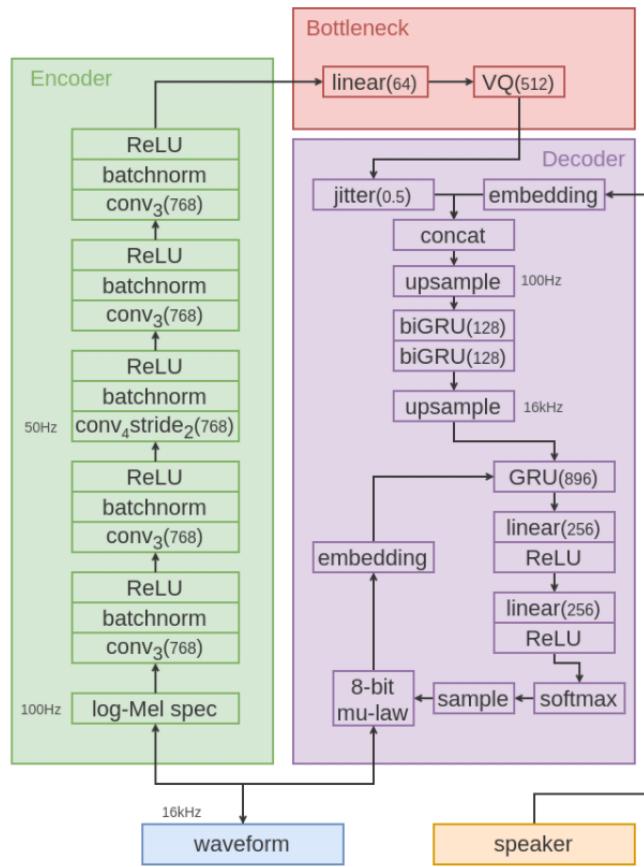


- Autoencoder(AE) : 데이터의 인코딩된 특징들을 학습한 뒤, 이 학습된 인코딩 표현에서 입력 데이터에 가깝게 생성하는 것을 목표로 하는 비지도 방식 인공신경망입니다. 이는 노이즈 제거나 이미지 압축/복원 등에 쓰입니다.

- Variational AE : Variational Autoencoder의 약자로, 기존의 Autoencoder는 어떤 데이터를 잘 압축하여 특징을 잘 뽑는 것을 목적으로 한 반면, VAE는 새로운 데이터를 생성하는 것이 목적(Generative model)인 발전된 Autoencoder입니다. 이 모델과 Autoencoder의 차이점은 AE는 잠재 공간 z 에 값을 저장하는 반면, VAE는 확률 분포를 저장하여 파라미터를 생성합니다.(변분추론이라는 기법 사용 – $q(z)$ 와 $p(z|x)$ 사이의 KL Divergence 값을 계산한 뒤 이 값이 줄어드는 쪽으로 $q(z)$ 를 갱신합니다.)

- VQ-VAE : Vector Quantized Variational Autoencoder의 약자로, VAE를 통해 생성한 이미지가 흐릿한 문제 해결을 위해 만들어졌습니다. 이미지를 인코더에 입력하고 출력인 잠재 변수의 벡터를 codebook(해당 인덱스와 관련된 벡터 목록)에 mapping하는 방식으로 구현합니다. VAE는 연속적 잠재 표현을 학습하는데 반해 VQ-VAE는 이산적인 잠재 표현을 학습합니다. 이는 현재 나와 있는 다른 알고리즘보다 다양하고 복잡한 데이터 분포를 더 잘 나타낼 수 있습니다.

- Zero-Speech Challenge 2020에서 사용된 VQ-VAE 모델 : 저희 팀이 주목한 기존 연구 모델은 VQ-VAE라는 모델을 이용합니다. 이 구조는 encoder와 decoder의 두 부분으로 이루어져 있는데, 학습된 표현들은 음성 콘텐츠만 포함하도록 조정되기 때문에 WaveNet decoder를 사용해서 인코더에서 손실된 정보들을 추론하도록 구성되어 있습니다.

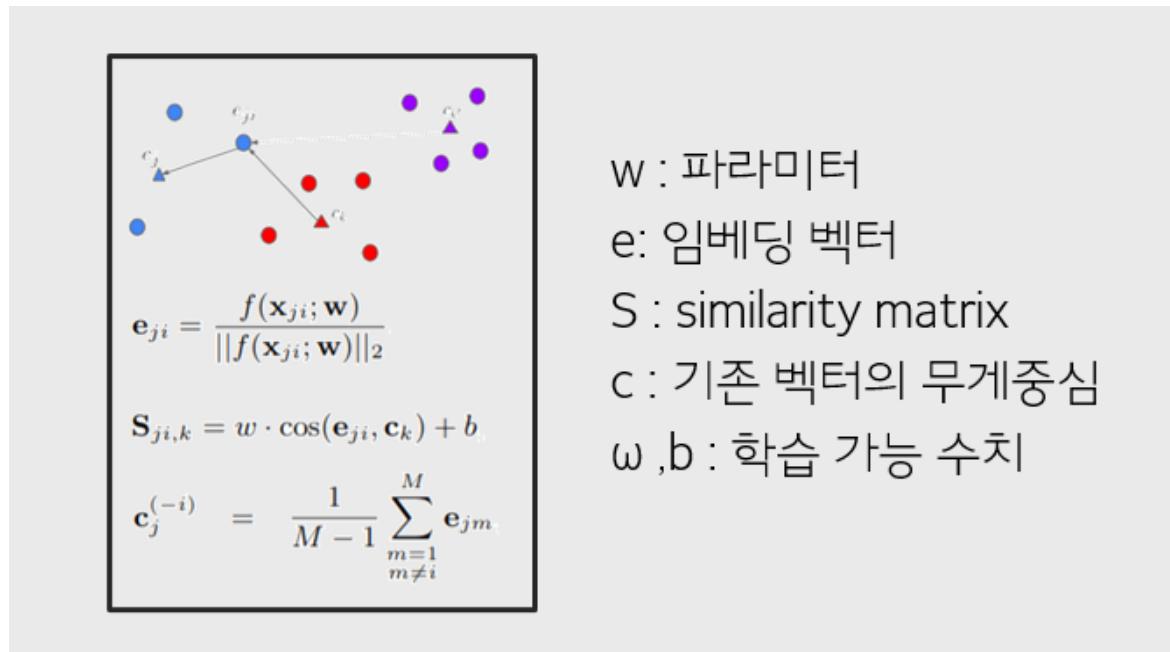


b. 기존 연구의 한계점

- 화자를 **one-hot** 벡터로 받기 때문에 화자 구분 능력이 떨어집니다.
- 학습할 때 사용하는 음성데이터 이외의 데이터에 대해서 합성 능력이 떨어집니다.

3. 제안하는 연구의 소개

a. Generalized End-to-End 모델



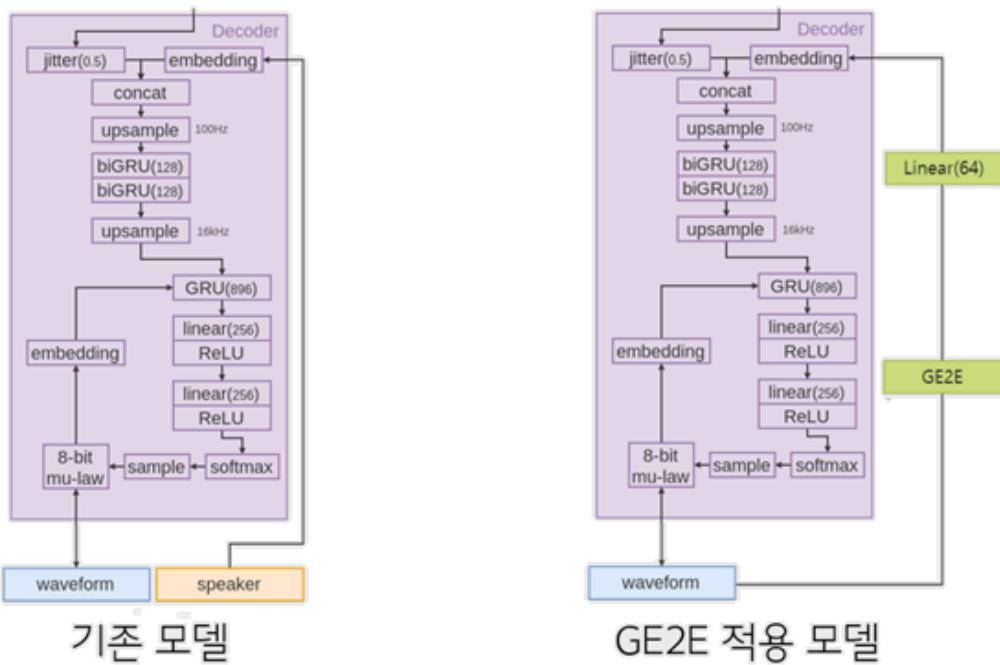
- GE2E는 화자간 목소리 차이를 분석하여, 목소리가 어떤 화자에 의해 발생된 것인지를 구분하는 최신 SV(Speaker Verification) 모델입니다.

- 이 모델은 우리가 사용하는 Zero Speech 모델과 유사한 기능을 수행하는 SV2TTS라는 모델에서 화자 정보를 기입하기 위해 쓰는 방법입니다.

- 이 GE2E라는 모델은 이전 TE2E 등의 화자 검증 모델과 달리 새로운 손실함수를 사용해 스피커 임베딩 벡터를 실제 스피커 벡터들의 무게중심으로 값에 가깝도록, 다른 스피커 벡터들의 무게중심으로부터 멀어지도록 training하므로 화자 검증 능력이 뛰어납니다.

4. 개선 방법론

a. Speaker Encoding을 활용한 모델 개선



- 현재 one-hot 벡터로 들어가서 화자 간 차이를 잘 반영하지 못하는 임베딩 레이어를 GE2E 모델을 활용해 더 유용한 정보를 반영하는 레이어로 교체합니다. 디코더가 화자간 차이를 더 잘 이해해 향상된 데이터를 생성할 것으로 기대됩니다.
- 결과물에서 화자간 차이를 더 잘 반영하는 것에 더해, 화자 임베딩 개선을 통해 전체 모델이 더 잘 학습할 것이라고 예상하였습니다.
- GE2E는 pretrained 된 Resemblyzer 모델의 weight를 고정하고 기존 임베딩과 같은 사이즈의 64D Linear 레이어에 연결했습니다.

b. 정규화를 통한 Overfitting 해결

- Weight Decay 적용으로 해결 : Overfitting 문제를 해결하기 위해 딥러닝에서 L2 정규화를 사용, weight값이 너무 커지지 않도록 오차 함수에서 weight이 커지는 경우에 대한 패널티를 부여해서 커지지 않도록 하였습니다.
- 기존 학습 방법으로는 일정한 시간의 학습 후 Gradient Overflow가 연속적으로 발생해서 오차가 급격히 발생하는 문제가 있었는데, 모델의 weight가 너무 커져서 mixed-precision training에 문제가 생긴 것으로 파악해 정규화를 진행하기로 하였습니다. 정규화를 통해 오버피팅을 막는 효과도 기대할 수 있습니다.

- 50만pt 학습 : Gradient Overflow문제를 제거하였습니다.

- 64만pt 학습 : 정규화 특성상 성능이 떨어질 수 있는 것을 감안해 충분히 학습된 모델을 정규화 제거 후 초과 학습했습니다.

5. 연구 방법

a. GE2E 화자 구분 성능 확인

- UMA 알고리즘

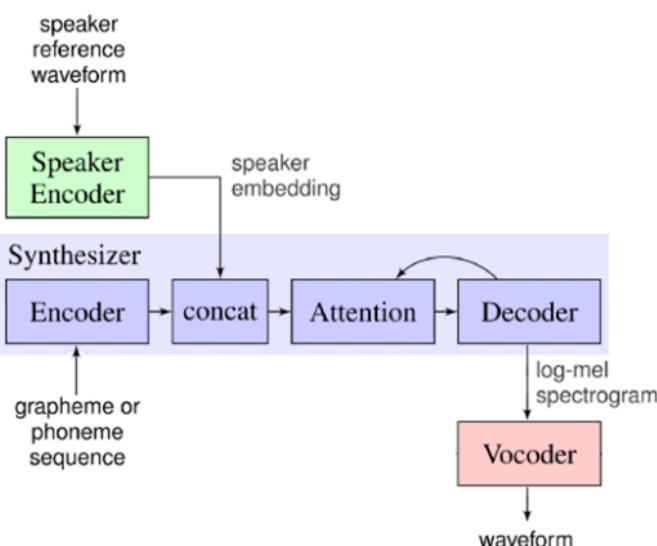
- 2차원 시각화

- 화자 유사도 계산

b. GE2E 적용

- GE2E 모델은, 우리 프로젝트와 유사하게 텍스트를 음성으로 변환시켜주는 SV2TTS라는 모델에서도 화자 정보를 추출하기 위해 이용하는데, 이 모델의 경우 GE2E의 힘으로, 짧은 목소리 파일만 있다면 그 사람의 목소리로 텍스트를 변환시키는게 가능합니다. 이 기술을 적용한 깃헙 프로젝트는, 3만개 가량의 깃헙 스타를 보유한 상당히 인지도가 높은 프로젝트입니다.

- 저희 기존 모델은 화자 정보를 one-hot 벡터로 받는데, 이 부분에 미리 train된 GE2E 모델의 출력이 들어갈 수 있도록 했습니다(정확히 말하면, 도식에 보이는 것처럼 기존 모델의 레이어 사이즈를 유지시키기 위해 64차원 벡터로 변환하는 fully-connected레이어도 추가했습니다). 미리 train된 GE2E 모델은 Resemblyzer라는 오픈소스 프로젝트를 사용했습니다.



c. 정규화

- Overfitting 문제를 해결하기 위해 쓰이는 기법으로, weight값이 커지지 않도록 막아줍니다.
오차 함수가 작아지는 방향으로만 학습 진행하면 특정 가중치 값들이 커지면서 오버플로우가 발생하는 등 결과가 나빠질 수 있습니다.
- Weight값이 너무 커지지 않도록 오차 함수에서 weight이 커지는 경우에 대한 패널티를 부여해서 커지지 않도록 합니다.

d. 새롭게 만든 모델로 전처리 및 트레이닝 수행

```
832 epoch:3499, recon loss:2.35E+00, vq loss:1.17E-02, perplexity:280.633
833 epoch:3500, recon loss:2.36E+00, vq loss:1.18E-02, perplexity:278.821
834 epoch:3501, recon loss:2.35E+00, vq loss:1.20E-02, perplexity:279.602
835 epoch:3502, recon loss:2.35E+00, vq loss:1.17E-02, perplexity:278.762
836 epoch:3503, recon loss:2.35E+00, vq loss:1.12E-02, perplexity:278.875
837 epoch:3504, recon loss:2.35E+00, vq loss:1.16E-02, perplexity:278.977
838 epoch:3505, recon loss:2.35E+00, vq loss:1.16E-02, perplexity:278.552
839 epoch:3506, recon loss:2.35E+00, vq loss:1.16E-02, perplexity:278.873
840 epoch:3507, recon loss:2.35E+00, vq loss:1.15E-02, perplexity:278.705
841 epoch:3508, recon loss:2.34E+00, vq loss:1.15E-02, perplexity:277.772
842 Gradient overflow. Skipping step, loss scaler 0 reducing loss scale to 131072.0
843 epoch:3509, recon loss:2.35E+00, vq loss:1.15E-02, perplexity:277.208
844 Gradient overflow. Skipping step, loss scaler 0 reducing loss scale to 65536.0
845 Gradient overflow. Skipping step, loss scaler 0 reducing loss scale to 32768.0
846 Gradient overflow. Skipping step, loss scaler 0 reducing loss scale to 16384.0
847 epoch:3510, recon loss:2.35E+00, vq loss:1.18E-02, perplexity:280.201
848 epoch:3511, recon loss:2.35E+00, vq loss:1.16E-02, perplexity:277.920
849 epoch:3512, recon loss:2.35E+00, vq loss:1.16E-02, perplexity:279.373
850 epoch:3513, recon loss:2.35E+00, vq loss:1.17E-02, perplexity:278.502
851 epoch:3514, recon loss:2.35E+00, vq loss:1.16E-02, perplexity:278.869
852 epoch:3515, recon loss:2.34E+00, vq loss:1.17E-02, perplexity:279.089
853 epoch:3516, recon loss:2.35E+00, vq loss:1.16E-02, perplexity:279.180
854 Saved checkpoint: model.ckpt-640000
855 epoch:3517, recon loss:2.35E+00, vq loss:1.16E-02, perplexity:278.998
856 epoch:3518, recon loss:2.36E+00, vq loss:1.16E-02, perplexity:278.999
857 epoch:3519, recon loss:2.35E+00, vq loss:1.15E-02, perplexity:277.183
858 epoch:3520, recon loss:2.35E+00, vq loss:1.15E-02, perplexity:278.379
859 epoch:3521, recon loss:2.35E+00, vq loss:1.16E-02, perplexity:279.724
860 epoch:3522, recon loss:2.34E+00, vq loss:1.15E-02, perplexity:278.540
861 epoch:3523, recon loss:2.35E+00, vq loss:1.17E-02, perplexity:279.177
862 epoch:3524, recon loss:2.35E+00, vq loss:1.16E-02, perplexity:278.712
863 epoch:3525, recon loss:2.34E+00, vq loss:1.17E-02, perplexity:278.313
864 epoch:3526, recon loss:2.35E+00, vq loss:1.16E-02, perplexity:277.271
865 epoch:3527, recon loss:2.34E+00, vq loss:1.17E-02, perplexity:279.649
866 epoch:3528, recon loss:2.35E+00, vq loss:1.16E-02, perplexity:277.114
867 epoch:3529, recon loss:2.35E+00, vq loss:1.16E-02, perplexity:278.038
868 epoch:3530, recon loss:2.35E+00, vq loss:1.16E-02, perplexity:278.497
869 epoch:3531, recon loss:2.35E+00, vq loss:1.16E-02, perplexity:279.582
870 epoch:3532, recon loss:2.35E+00, vq loss:1.17E-02, perplexity:279.627
```

e. 모델 평가

- reconstruction loss

- perplexity

- vq-loss

- abx score

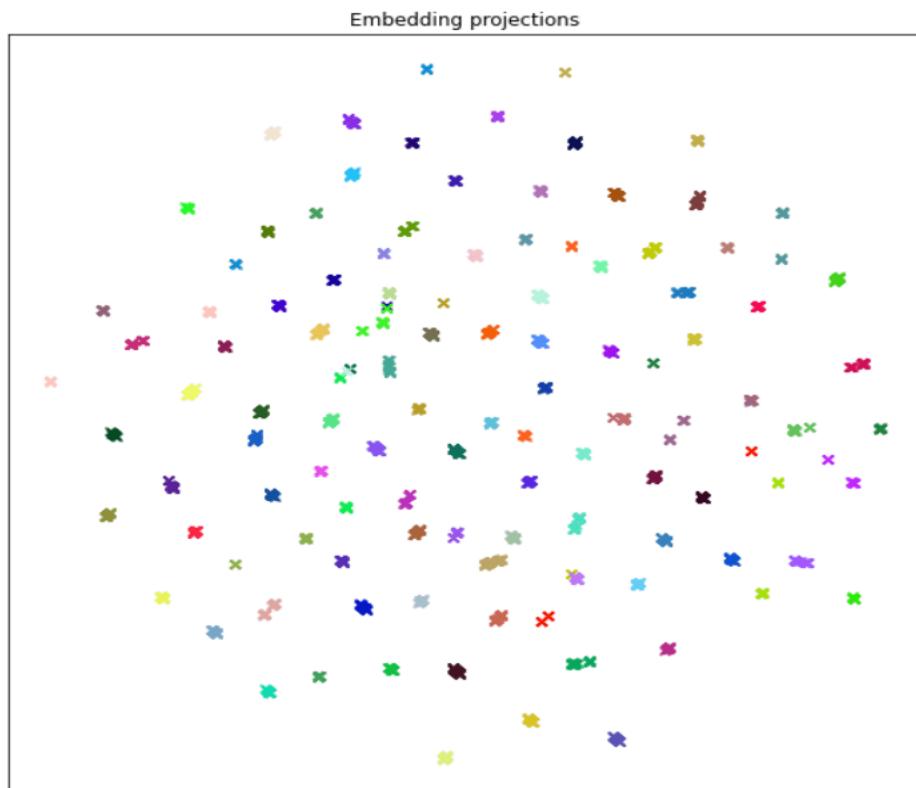
- bitrate

- MOS

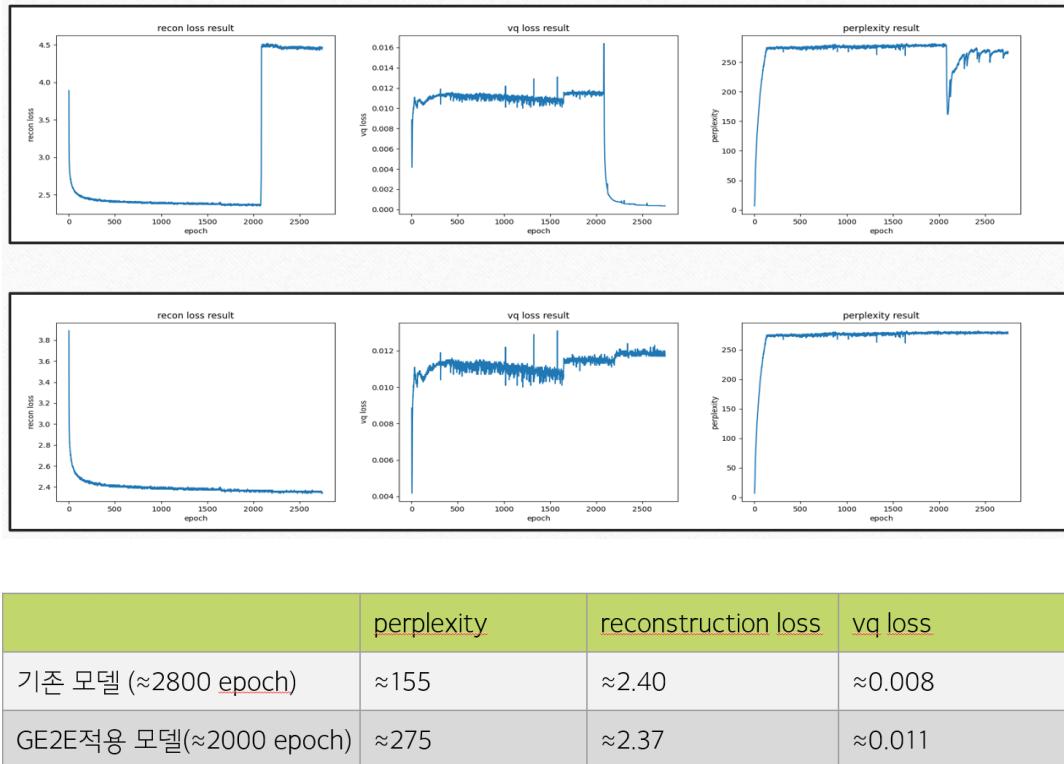
6. 연구 결과

a. GE2E 화자 구분 성능 확인

- 다음은 미리 train된 ge2e 모델의 성능을 평가한 결과입니다. 그림은 UMA 알고리즘을 이용하여 화자벡터들을 2차원 상에 투사한 것으로 같은 색의 점은 같은 화자입니다. 같은 색의 점들이 모여있는 경향을 확인 할 수 있습니다.



b. 트레이닝 로그 데이터 분석



- perplexity : 분기계수의 일종으로 낮을 수록 정확도가 높음
- reconstruction loss : 재생성된 waveform과 기존 waveform 비교
- vq loss : VQ-VAE에서 발생하는 손실

c. ABX score, bitrate 비교

c-1. 새로운 모델로 변환한 음성을 이용하여 ABX score와 bitrate를 계산한 결과

```
{  
    "2019": {  
        "english": {  
            "scores": {  
                "abx": 13.946426720951166,  
                "bitrate": 417.91478652705155  
            },  
            "details_bitrate": {  
                "test": 417.91478652705155  
            },  
            "details_abx": {  
                "test": {  
                    "cosine": 13.946426720951166,  
                    "KL": 50.0,  
                    "levenshtein": 35.40295181899956  
                }  
            }  
        }  
    }  
}
```

c-2. 기존 모델과의 비교

	기존 모델(50만pt)	새로운 모델(50만pt)
ABX Score	14.04	13.95
Bitrate	412.24	417.91

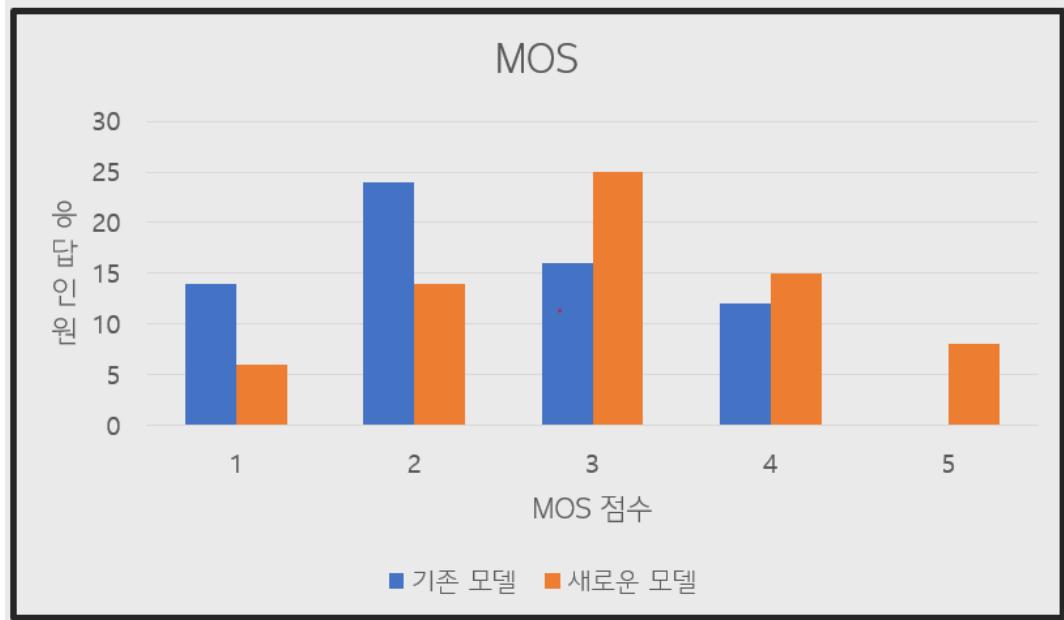
- ABX score : 최대한 작은 언어단위로 분리해 놓은 기존의 데이터와 거리를 측정하여

음성의 완성도를 측정하는 방법

- Bitrate : 테스트 동안 전송된 비트의 수

d. MOS 평가 비교

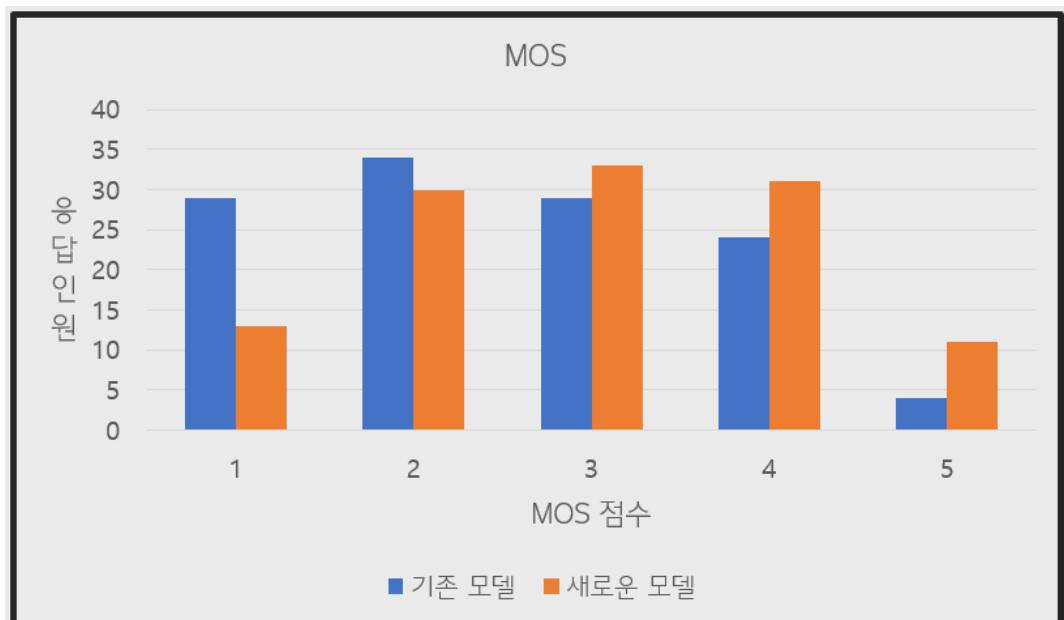
d-1. 데이터셋 내부 음성의 변환 결과



기존 모델 평균 : 2.39

새로운 모델 평균 : 3.08

d-2 데이터셋 외부 음성의 변환 결과



기존 모델 평균 : 2.50

새로운 모델 평균 : 3.02

e. MOS 평가 결과 통계적 분석

두 개의 MOS 테스트 결과에 대해 통계적 분석을 진행했습니다.

e-1 데이터셋 내부 음성의 변환 결과

제로스피치 데이터셋에 있는 세 음성 샘플을 구모델과 신모델로 변환한 결과에 대한 MOS점수를 Repeated Measures ANOVA 기법을 사용해 신모델에 대한 점수의 통계적 유의미성을 파악했습니다. Within subjects 독립변수로 구모델과 신모델, between subjects 독립변수로 변환된 음성 샘플을 설정하고, 종속 변수는 MOS 점수로 설정했습니다. α 는 0.005로 설정합니다. 표 S1-1에서 descriptive statistics를 확인할 수 있습니다. 표 S1-2에서 sphericity assumption을 테스트한 결과를 확인합니다. p-value가 모두 0.005보다 큰 값을 가져서 sphericity assumption을 만족합니다. Repeated Measures ANOVA 테스트에서, 표 S1-3의 첫 번째 행에서 모델의 차이에 의한 p-value가 0.003으로 α 보다 작습니다. 따라서 저희 모델(신모델)에 대한 MOS점수가 통계적으로 유의미하다고 볼 수 있습니다. 도식 S1-1은 음성 샘플 1, 2, 3의 구모델과 신모델로의 변환결과에 대한 MOS점수의 평균을 도시한 것입니다.

모델	음성	Mean	SD	N
구모델	1	2.304	0.926	23
	2	2.522	1.082	23
	3	2.435	1.080	23
신모델	1	2.522	1.163	23
	2	3.261	1.010	23
	3	3.478	1.039	23

[표 S1-1]

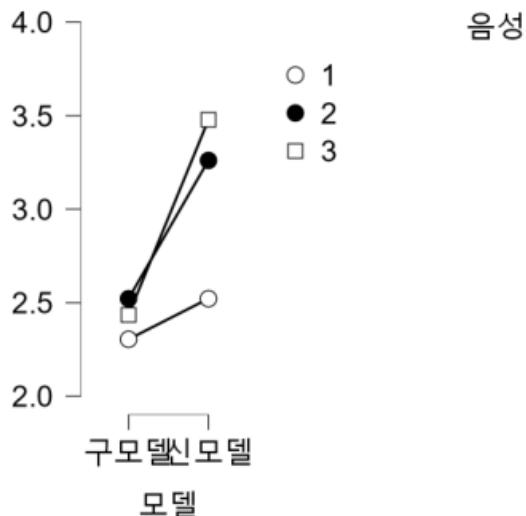
	Mauchly's W	Approx. χ^2	df	p-value	Greenhouse-Geisser ϵ	Huynh-Feldt ϵ	Lower Bound ϵ
음성	0.770	5.501	2	0.064	0.813	0.868	0.500
모델 * 음성	0.970	0.630	2	0.730	0.971	1.000	0.500

[표 S1-2] Test of Sphericity

Cases	Sum of Squares	df	Mean Square	F	p	η^2
모델	15.333	1	15.333	11.500	0.003	0.126
Residuals	29.333	22	1.333			
음성	8.101	2	4.051	5.707	0.006	0.067
Residuals	31.232	44	0.710			
모델 * 음성	4.014	2	2.007	2.651	0.082	0.033
Residuals	33.319	44	0.757			

Note. Type III Sum of Squares

[표 S1-3] Within Subjects Effects



[도식 S1-1]

e-2 데이터셋 외부 음성의 변환 결과

생소한 화자의 다섯 음성 샘플을 구모델과 신모델로 변환한 결과에 대한 MOS평균 점수를 **Repeated Measures ANOVA**기법을 사용해 신모델에 대한 점수의 통계적 유의미성을 파악했습니다. **Within subjects** 독립변수로 구모델과 신모델, **between subjects** 독립변수로 음성 샘플의 소스 화자를 설정하고, 종속 변수는 MOS 점수로 설정했습니다. α 는 0.005로 설정합니다. 표 S2-1에서 **descriptive statistics**를 확인할 수 있습니다. **Repeated Measures ANOVA** 테스트에서, 표 S2-2의 첫 번째 행에서 모델의 차이에 의한 **p-value**가 0.001 미만의 으로 α 보다 작습니다. 따라서 저희 모델(신모델)에 대한 MOS점수가 통계적으로 유의미하다고 볼 수 있습니다. 도식 S2-1은 각 소스 화자에 대한 구모델과 신모델로의 변환결과에 대한 MOS점수의 평균을 도시한 것입니다.

모델	타겟 화자	소스 화자	Mean	SD	N
구모델	1	Biden	1.667	0.748	144
		Game	3.250	1.238	144
		Hillary	3.000	1.084	144
		Thunberg	2.417	1.191	144
	2	Trump	2.250	0.832	144
신모델	1	Biden	1.833	0.901	144
		Game	2.500	0.961	144
		Hillary	3.500	0.869	144
		Thunberg	2.417	1.191	144
	2	Trump	2.167	0.989	144

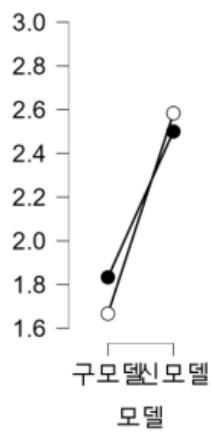
[표 S2-1]

Cases	Sum of Squares	df	Mean Square	F	p	η^2
모델	186.050	1	186.050	315.040	< .001	0.046
모델 * 소스 화자	24.700	4	6.175	10.456	< .001	0.006
Residuals	422.250	715	0.591			
타겟 화자	1.250	1	1.250	1.735	0.188	3.059e -4
타겟 화자 * 소스 화자	80.500	4	20.125	27.927	< .001	0.020
Residuals	515.250	715	0.721			
모델 * 타겟 화자	4.050	1	4.050	9.456	0.002	9.912e -4
모델 * 타겟 화자 * 소스 화자	10.700	4	2.675	6.245	< .001	0.003
Residuals	306.250	715	0.428			

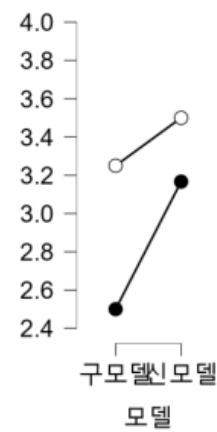
Note. Type III Sum of Squares

[표 S2-2] Within Subjects Effects

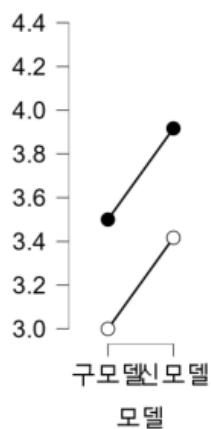
소스 화자: Biden



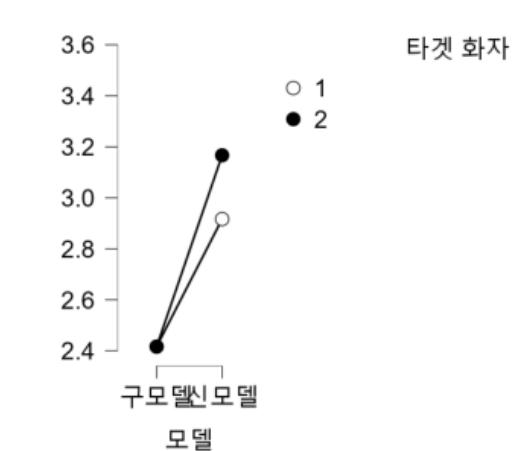
소스 화자: Game



소스 화자: Hillary

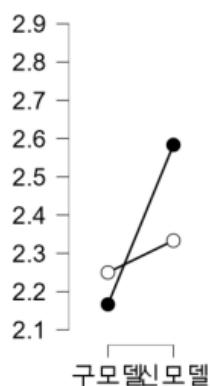


타겟 화자



타겟 화자

소스 화자: Trump



타겟 화자

[도식 S2-1]

f. 결론

-GE2E 기법을 기존 모델에 도입하고 Weight Decay를 설정하여 새로운 모델을 만든 결과, 음성 변환 능력이 기존 모델보다 향상되었습니다.

-성능 향상 원리 : GE2E 기법을 통해서 화자 구분 능력 향상, 정규화를 통해 Gradient Overflow 문제를 해결하였습니다.

7. 팀의 구성 및 역할

신명진 : GE2E 연구 방향 아이디어 제시, GE2E 화자 구분 성능 확인, weight-decay 이용한 정규화 설정, 전시회 영상 short version 제작, MOS 결과 통계적 기법 활용 분석, 질문답변

위준복 : 연구제안발표, 새로운 모델로 트레이닝 및 음성 변환, 기존 테스트 데이터 MOS 수행, 새로운 테스트 데이터 MOS 수행, 최종보고서 작성

조형진 : 연구제안발표 ppt 제작, 트레이닝 로그파일 분석, 중간발표 ppt 제작, ABX-Test 수행, 중간발표, 최종발표

공통 : GE2E 적용한 새로운 모델 구현, 연구 진행상황 보고 및 연구지도확인서 작성(1~6), 중간보고서 작성, 최종발표 ppt 제작, 전시회 ppt 제작, 전시회 영상 long version 제작

8. 참고 문헌

- 기존 모델: Chorowski, Jan, et al. "Unsupervised speech representation learning using wavenet autoencoders."
- 인코더(VQ-VAE): van den Oord, Aaron, and Oriol Vinyals. "Neural discrete representation learning."
- 디코더(WaveNet): Google, "WaveNet: A Generative Model for Raw Audio"
- SV2TTS: "Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis"
- Mixed Precision Training: Paulius Micikevicius et al. "Mixed Precision Training"
- Regularization: Jan Kukacka et al. "Regularization for Deep Learning: A Taxonomy"
- Repeated Measures ANOVA:
<https://statistics.laerd.com/statistical-guides/repeated-measures-anova-statistical-guide.php>