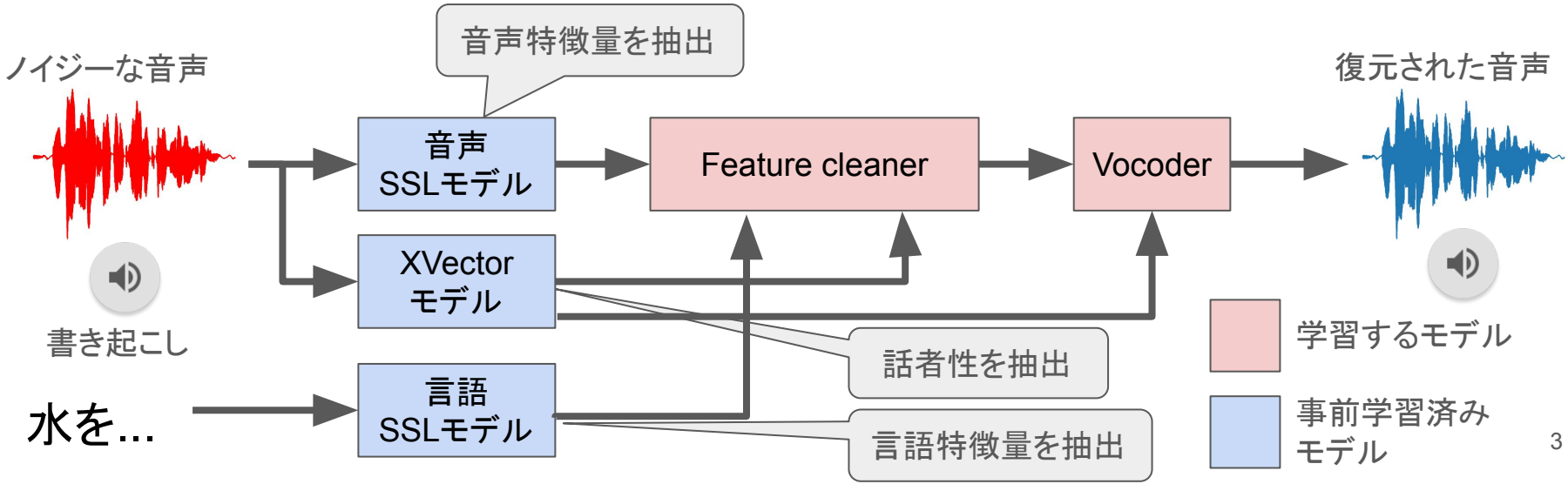
Miipherを実装し評価しました

東大　猿渡・高道研究室
中田亘

目次

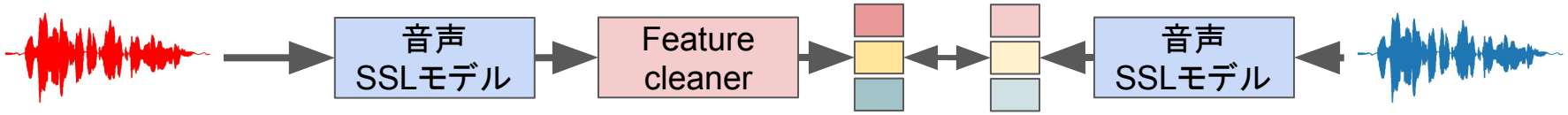
Miipherとは

- 音声をSSLモデル特徴量空間で復元する技術
- 高い性能を示しており、大規模TTSコーパス構築に有用
  - LibriTTS-R[Koizumi+23]

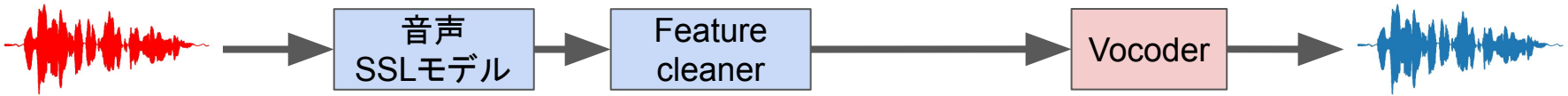


Miipherの学習（XVector，言語SSLモデルは省略）

Stage1(a): ボコーダの学習



Stage1(b): Feature Cleanerの学習



Stage2: ボコーダのファインチューニング

自分の実装とPaperの主な相違点

	論文	自分の実装
データセット	Google社内データ 670時間	LibriTTS-R[Koizumi+23] 585時間 JVS corpus[Takamichi+20] 24時間
音声SSLモデル	W2v-BERT[Chung+21]	WavLM[Chen+21]
言語SSLモデル	PnG BERT[Jia+21]	XPhoneBERT[Nguyen+23]
Feature cleanerの主たる構造	DF-Conformer[Koizumi+21]	Conformer[Gulati+20]
ボコーダ	WaveFit[Koizumi+22]	HiFi-GAN[Kong+20]

全て Closed source

評価1: 復元音声の客観的評価

評価に使用したデータセット
- JSUT[Takamichi+20] BASIC5000 0001~0100 日本語　女性話者 1名
- CMU ARCTIC[Kominek+2003]　各話者100発話　男性2名 女性2名

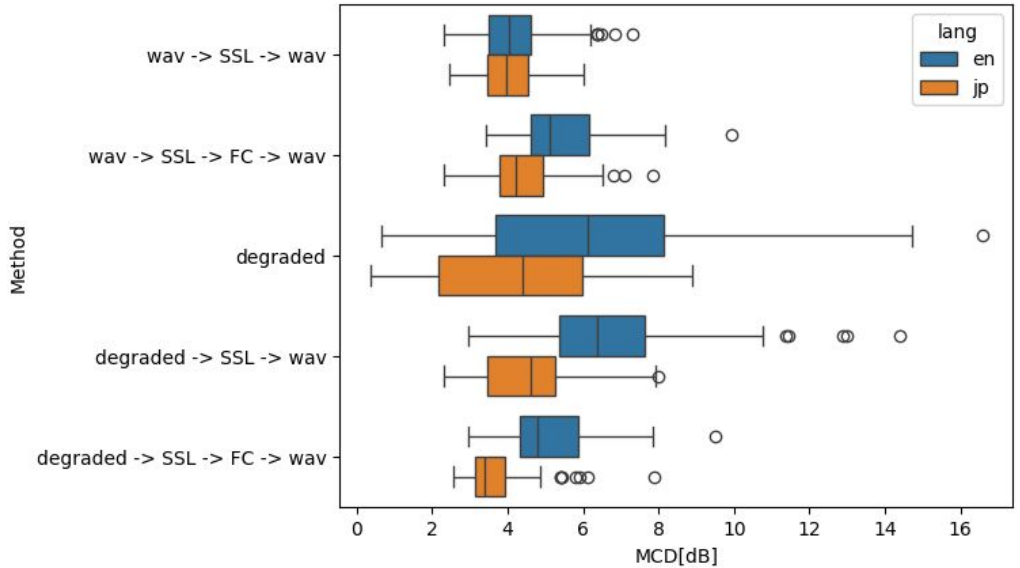
評価指標
- 音声の品質：Mel-cepstrum distortion (MCD)
- 話者性の保存：X-vector コサイン類似度
- 言語情報の保存：ASR結果の文字誤り率
- 韻律の保存：Log F0 RMSE

原音声劣化手法
- Codecの適用（mp3, vorbisなど）
- Reverbを50%の確率で適用
- 背景雑音，音楽の適用　SNR 30dBから5dBを一様分布からサンプリングして適用

比較手法

	人工的な劣化	音声SSL特徴量	Feature cleaner
wav （原音声そのまま）			
wav -> SSL -> wav		✅	
wav -> SSL -> FC -> wav		✅	✅
degraded	✅		
degraded -> SSL -> wav	✅	✅	
degraded -> SSL -> FC -> wav	✅	✅	✅

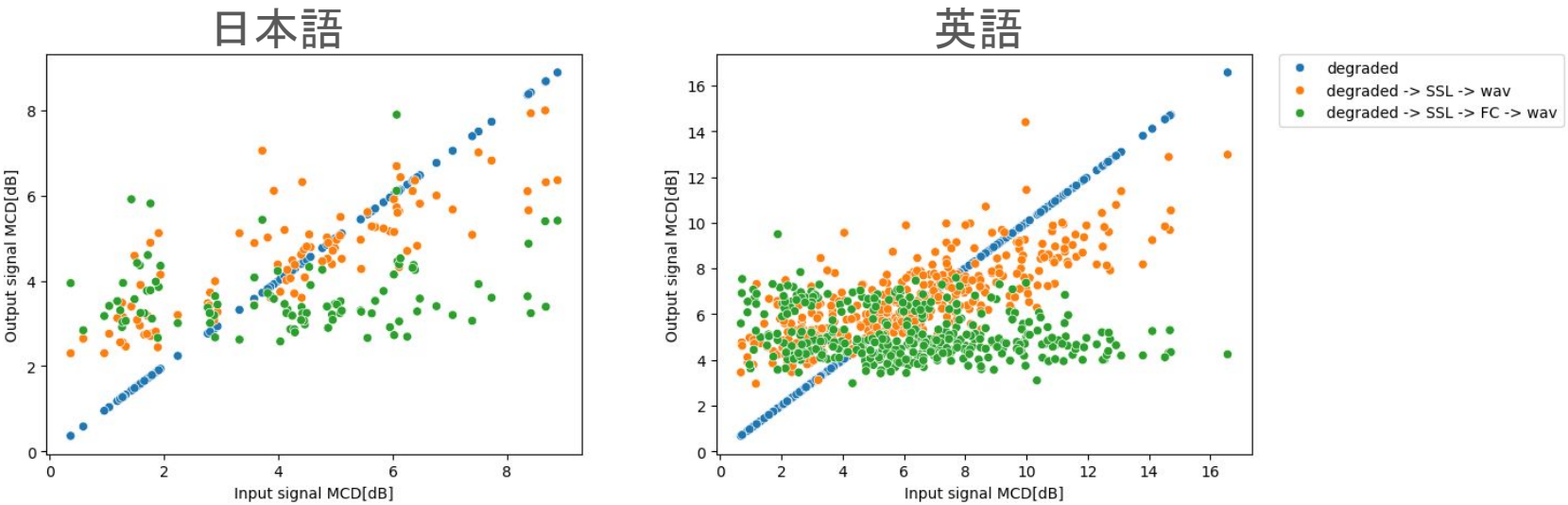
メルケプストラム歪み（MCD）の評価結果



clean音声入力の場合　-> SSL特徴量、Miipher使用により劣化
Degraded音声入力の場合 -> SSL特徴量、Miipherの使用により分布が狭くなる
日本語の方がMCDが良い値

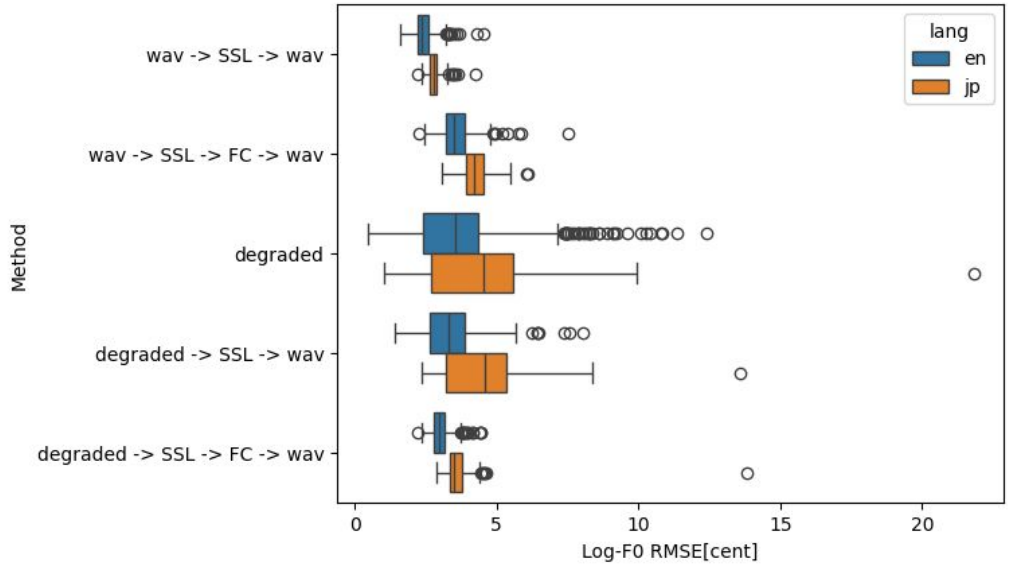
- 英語データセット：LibriTTS-R（Google Miipherで復元された音声）日本語データセット：JVSコーパス

Degraded音声のMCDと各手法のMCD比較



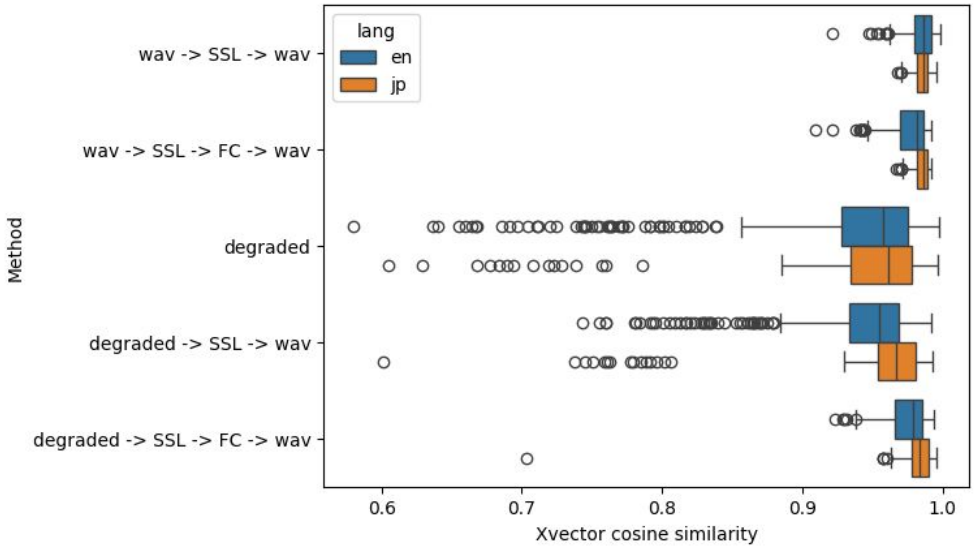
- SSL特徴量を使用することでMCDの増加が改善
- Miipherを使用することでMCDの増加が大きく改善
- 入力音声がcleanに近い場合逆に劣化が発生

Log-F0 RMSEの評価結果



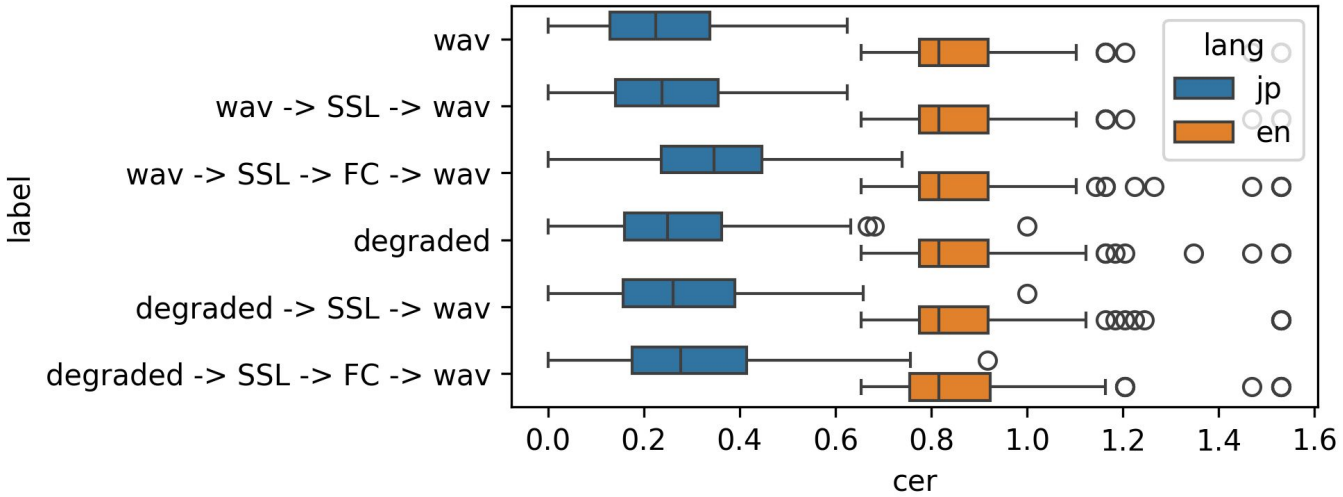
- 日本語ではLog-F0 RMSEが劣化
  - 使用したSSLモデルが英語事前学習済みモデルであるから？
- Degraded音声に対してはSSL特徴量，FCの利用により改善

X-Vectorコサイン類似度（話者性）の評価結果



- clean音声入力の場合　-> SSL特徴量、Miipher使用により劣化
- Degraded音声入力の場合 -> SSL特徴量、Miipherの使用により改善
- X-Vectorが劣化音声に対してロバストではない

文字誤り率



- 英語ではCERの劣化は少ない．むしろFCを使用することでCERが少し改善
  - 音声SSLモデルが英語で事前学習していることが要因？
- 日本語では，SSL特徴量やFCを使用することでCERが劣化
  - F0のエラーが大きいのが一因と思われる

評価2 劣化の種類による復元音声の品質の変化

先ほどと同じデータセットを以下の条件で比較

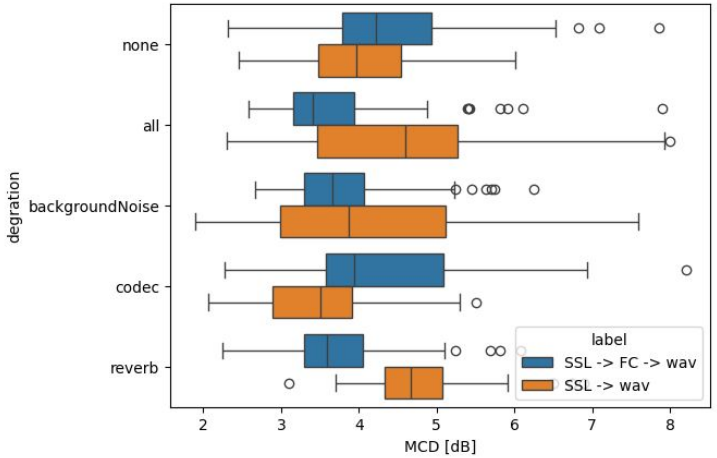
	残響の適用	コーデック劣化の適用 （mp3, vorbisなど）	背景雑音，音楽の適用
none			
Reverb	✅		
Codec		✅	
Background Noise			✅
All	✅	✅	✅

評価指標

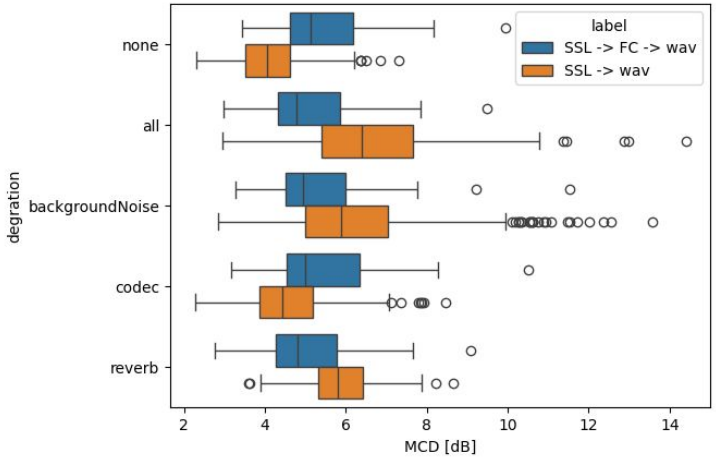
- MCD, X-Vectorコサイン類似度，Log-F0 RMSE

劣化の種類によるMCDの変化

日本語



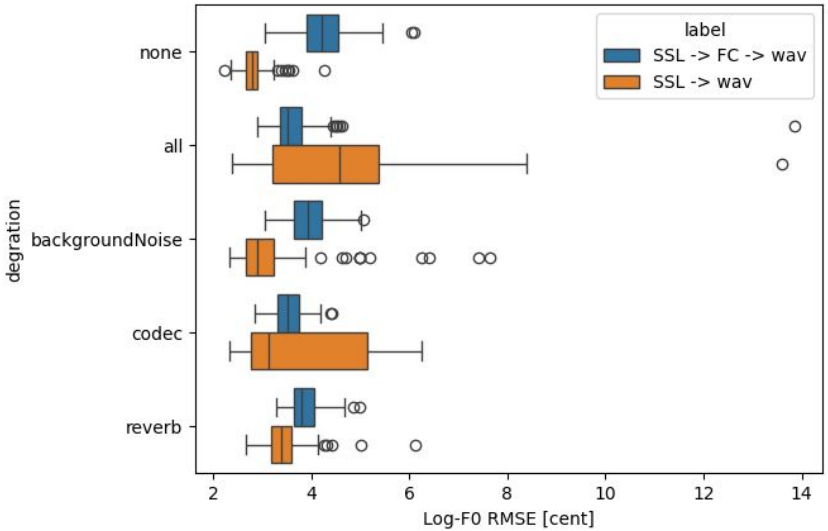
英語



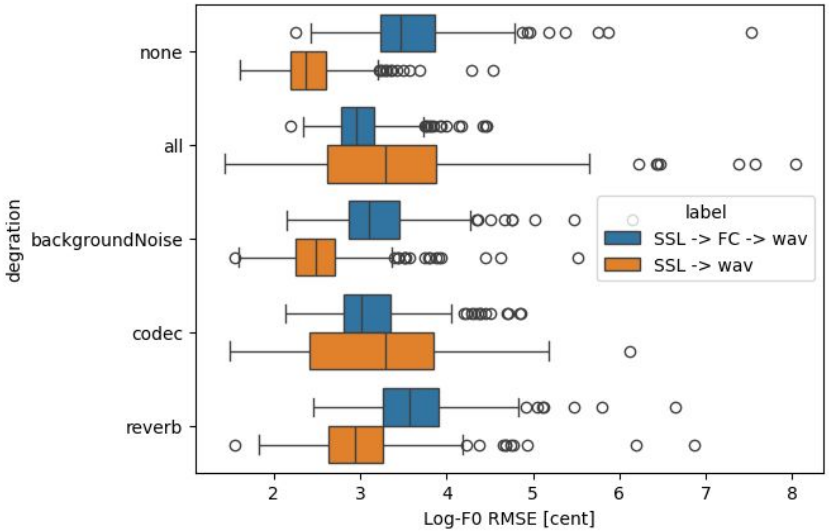
- 言語に寄らず，全体的な傾向は一致
- CodecにMiipherはロバストではない

劣化の種類によるLog-F0 RMSEの変化

日本語



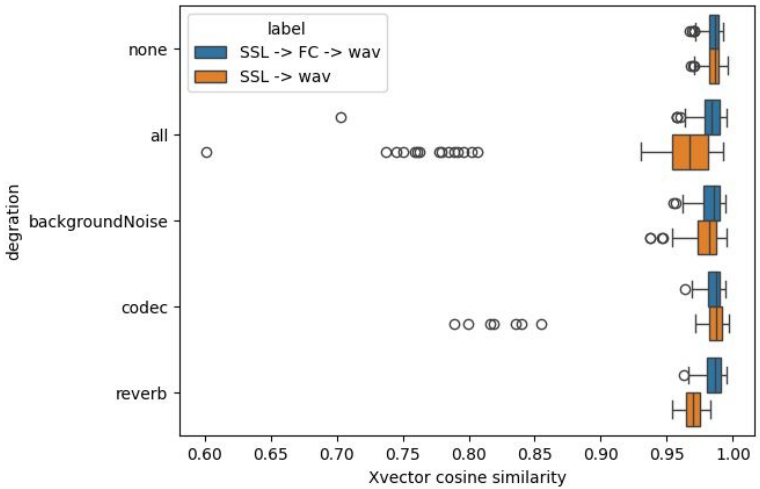
英語



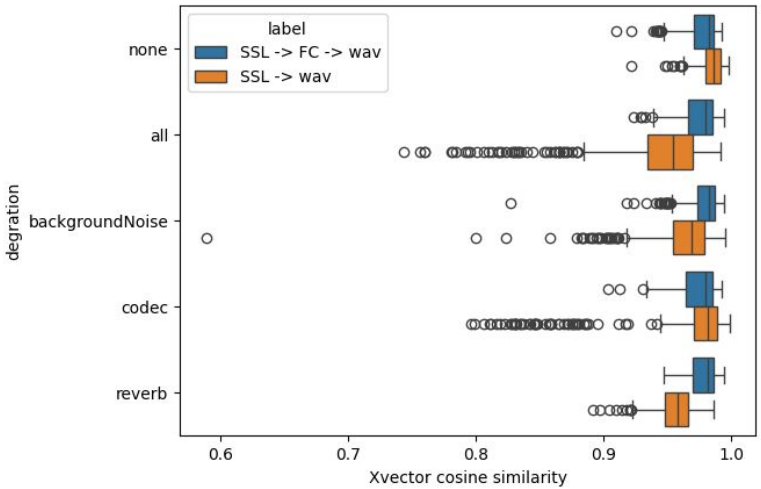
- F0に関してはbackground noise，reverbにおいて性能が劣化
- 言語に寄らず同様の傾向

劣化の種類によるXVector（話者性）の変化

日本語



英語



- XVectorに関しては特定の劣化に対して弱いなどは確認できない

自分が実装したMiipherの考えうる用途と改善方針

残念ながら自分の実装ではスタジオ品質は出来ていない

用途: Cleanである必要は無いが，音声のみ含まれていて欲しい学習
- GSLMなどの音声言語モデル学習データの前処理
- TTSモデルの事前学習

改善方針

- 真のCleanデータをより多く集める （LibriTTS-RはGoogleのMiipherにより復元された音声）
- Clean音声を多く学習時に入力（今はほぼ存在しない）
- ノイズロバストなX-Vectorを学習
- F0-awareな音声SSLモデルの設計 or ボコーダをF0で条件付け
- 日本語学習ずみ音声SSLモデルの使用
- 複数のそうのFusion

まとめ

実装したMiipherの性能を評価

復元音声の品質

- 😁Degraded音声に対してMiipherを使用する事により改善
- 😨clean音声に対して適用すると劣化

劣化手法に対するロバスト性

- MCD: Codec劣化に対して弱い
- F0: 背景雑音，反響に対して弱い