声色情報と韻律特徴を用いたマルチファクターな音声仮名化

Multi-factor Speech Pseudonymization Using Speaker Characteristic and Speech Rhythm-Based Speaker Embedding

伊藤葵 (Aoi Ito)

法政大学 情報科学部 ディジタルメディア学科
aoi.ito.8q@stu.hosei.ac.jp

Abstract

Pseudonymized speech data make it possible to effectively utilize the information contained in the speech data(content, emotion, etc.) while protecting the speaker's privacy. One method for protecting privacy is speech anonymization using x-vectors that embed one's spectral features. However, since humans also use prosodic information to distinguish between speakers, there is a risk that some people will guess the speaker from the prosodic information by converting only speaker features. In this paper, along with converting the x-vector as speaker features, we also propose pseudonymizing prosodic information by converting each phoneme in the utterance and its duration. Furthermore, in this experiment, in order to evaluate the performance of Baseline System 1 on the VoicePrivacy Challenge, we used a Japanese speech dataset instead of the traditional English corpus and analyzed the distribution of x-vectors.

1 はじめに

音声は, 声色, 音高, 韻律など様々な情報が含まれており, それらの情報を分析することで何を話しているかという発話内容といった言語情報, 誰が話しているかという話者情報, 話し方等から伝わってくるパラ言語情報など様々な情報を得られる. 近年, 音声認識や話者照合, 音声を用いた感情認識などの研究が進むにあたり, 音声内に含まれるこれらの情報を活用したサービス (自動議事録作成アプリ, スマートスピーカーなど) が企業や一般家庭, 医療現場といった幅広い分野で導入されている. 音声には人々の活動の幅を広げるための情報を多く含有している一方で, 用途とは関係ない情報から予期せぬプライバシー侵害のリスクにつながる恐れがある. 例えば, 音声認識タスクにおいて, 対象とする音声内の情報は発話内容である. しかし, 同時に音声から読み取れる声色情報, 音高, 話し方といった情報から, 音声

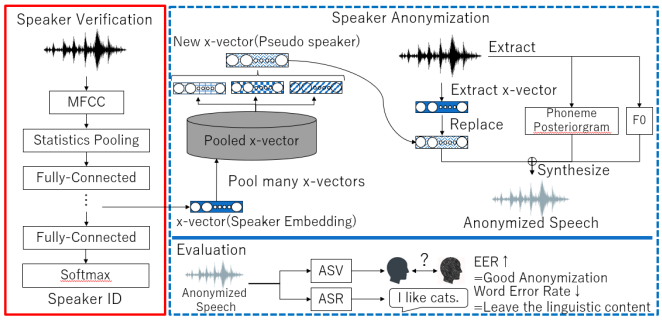

Fig.1: Overview of Previous Speaker Anonymization using x-vector. This method is proposed by Fang et al.

認識タスクに関係のない話者推定につながる可能性がある. また, 研究用途やサービス提供において企業をはじめとする各機関が音声データを保管する場合がある. 音声を生データのまま保管することで, 悪意のある第三者が音声データから話者, 性別等の個人情報を分析, 悪用するといったプライバシー侵害のリスクがある. このように, 生の音声データは必要な情報以外に話者自身の特徴 (話者性) に関する情報も保持しており, General Data Protection Regulation でも音声データの保護が言及されていることから, 音声データに含まれる個人情報及び音声データの扱い方について注目が寄せられている. 今後音声を活用する場合は, 音声データの特徴を踏まえたうえで, 音声データ内の個人情報を保護する処理が不可欠である.

音声データに対するプライバシー保護について, VoicePrivacy Challenge [1] コンペティションをはじめ, 音声匿名化 (仮名化)[2, 3] という研究が盛んとなっている. VoicePrivacy Challenge で取り扱っている音声匿名化では, リテラルレベルの発話内容を保持しつつ, 話者の特定に繋がる情報を処理 (マスキング, 平均化, 除去など) することで, 話者のプライバシー保護を目指している. 特に, 話者を匿名化するだけでなく匿名化後も発話内容をリテラルレベルで保持することにより, 話者のプライバシーを保護したまま, 音声データを音声認識等のシステムに活用できる. このコンペティションでは, 声色情報を用いた匿名化手法をベースラインシステムの一つとして公開している. 匿名化

Supervisor: Prof. Katunobu Itou

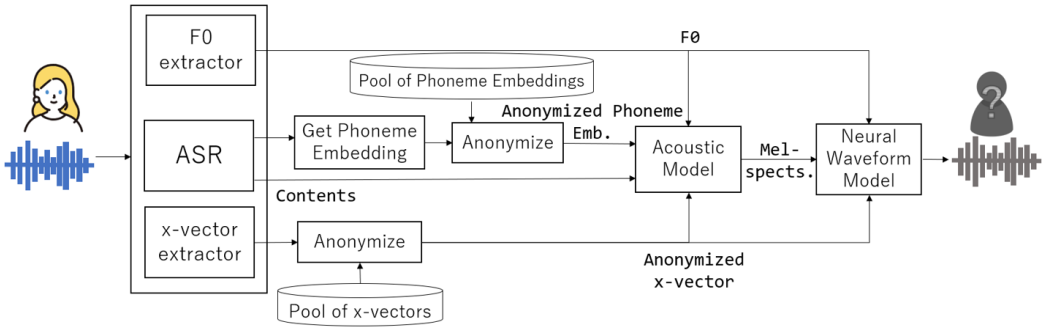

Fig.2: Overview of Proposed Speaker Pseudonymization Using Phoneme Embedding

済み音声に対する発話内容の保持レベルを評価する際は音声認識を用いており, 本コンペティションでは英語の音声認識タスクにて評価している.

本稿では, VoicePrivacy Challenge で公開されているベースラインシステムについて日本語コーパスを用いて評価し, 従来研究において取り扱ってきた話者特徴量の埋込である x-vector[4] を用いた声色情報の変換に加え, 話者推定の手がかりとなる韻律情報を明示的に扱い変換することによる, マルチファクターな音声データの仮名化手法を提案する.

2 従来研究

2.1 音声匿名化・仮名化

匿名化とは, 個人情報の復元が出来ないよう個人の特定に繋がる情報を加工することであり, 音声匿名化では, 話者性を音声内の言語的およびパラ言語的側面から切り離す. すなわち, 音声匿名化によって, **誰が**, **それを**話したかに関する情報がスピーチから削除されるが, **何が**, **どのように**話されたかは維持される. ただし, 音声匿名化は, 発話の言語的内容が保存されるだけであり, その他は韻律的側面の一部が保存されるだけである. そのため, 音声匿名化のアプローチでは, 発話を通じて話者が表現した感情, 話者の会話スキルや病理学的状態などに応じた明瞭度の変化など, 目的に応じて関心のあるパラ言語的特徴を保持・有効活用できない可能性がある. この目的に応じたデータの有用性という観点に対し考案された手法が, 音声仮名化というアプローチである. 仮名化とは, 可逆的な匿名化であり, 音声仮名化は音声コーパス共有によるプライバシー侵害のリスクと音声データそのものから得られる利益 (言語情報, 韻律情報, パラ言語情報など) のバランスを調整し, より多くの活用に繋げられる手法である. 音声仮名化は, 話者の秘匿化に加え, 匿名化中に隠蔽された追加情報を用いデータを再識別できることを前提としているため, 音声匿名化と比較しより実用的なアプローチといえる. このように, 音声仮名化は, 第三者による意図しない再識別のリスクと, データとしての有用性との間におけるトレードオフを考慮し, 用途に沿ってプライバシー上の利点を享受できるよう仮名化する必要がある [5].

以上の音声匿名化・仮名化について, 音声匿名化・仮名化手法の研究促進を目的に, 2020 年から VoicePrivacy Challenge [1] というコンペティションが開催されている. 音声から抽出できる特徴量 x-vector [4] を用いた深層学習に基づく音声匿名化や McAdams 係数 [6] を用いた音声匿名化手法が VoicePrivacy Challenge のベースラインシステムとして公開されており, 多くの研究はこれらのシステムを提案手法のベースや評価時の一基準として利用している. 最新の試みでは, 話者匿名化モデルに対する攻撃モデルの開発に焦点が当てられた VoicePrivacy Attacker Challenge が実施されている.

2.2 x-vector を用いた音声匿名化

VoicePrivacy Challenge[1] で紹介されているベースラインシステムの一つに, x-vector を用いた手法 [2] がある. 概要を図 1 に示す.

はじめに, 発話から発話内容と話者識別に用いる話者特徴量 (x-vector) を抽出する. x-vector [4] とは, 効率的に時間構造の情報を集約して表現するモデル Time Delay Neural Network（TDNN）を用いて抽出される話者埋め込み (Speaker Embedding) である. TDNN を用いて学習された話者認識モデルにおいて, 分類層の直前には話者情報が凝縮されていると考え, 分類層の直前の層の出力を話者性の埋め込みとして抽出する方法である. このように抽出された x-vector は, スペクトル空間内で多数の話者が均等に分類できるよう, 512 次元まで次元を落とした特徴量である. 抽出した x-vector に対し, [7] は事前訓練済みの変換モデルを用いて, 非識別化を試みた. これに対し [2] は, 一つの変換関数のみを学習し, かつ複数の x-vector をランダムに選択し平均を取ることで生成した疑似話者の特徴量を用いる匿名化を提案した.

[2] は, 音声のリテラルレベルでの発話内容を保ったまま話者照合システムの等価誤り率 (Equal Error Rate: EER) を上昇, すなわち匿名化性能を向上させた.

2.3 発話リズムの埋込

[8, 9] は, 音声合成において, 個人ごとの音素継続時間長のモデル化に適した話者埋め込み手法を提案した. 従来, 学習時に使用されてきた x-vector やメルスペクトログラム (人間の聴覚に基づいたメル尺度を周波数軸にしたスペクトログラム) といった特徴量は, 発話リズムといった時間特徴量を明示的に扱っていな

い．そこで, [8] は, 音素とその継続時間長を用いて発話リズムを埋め込むことにより, 時間的特徴量に基づく話者埋め込みベクトルの生成手法を提案した．x-vector では MFCC(Mel-Frequency Cepstrum Coefficient: メル周波数ケプストラム係数) という人の聴覚特性を考慮しながらスペクトルの概形を表現した特徴量を入力として話者認識モデルを学習, 埋め込みを抽出するのに対し, [8] ではまず入力音声に対して音素レベルのアライメントを取得し, 音素とその継続時間長を取得する．ここで, RNN(Recurrent Neural Network) のように, ある中間層で計算した情報を, 再び中間層の入力として繰り返し処理を行うネットワーク構造 (GRU(Gated Recurrent Unit), LSTM(Long Short Term Memory)) に代わって, Transformer の Encoder 部を利用する．Transformer Encoder に対し, 音素の onehot 表現と継続時間長 (1 次元のベクトル) を concat し, 前後数音素でまとめた系列を入力として話者認識モデルを学習する．Transformer Encoder を用いて, 話者性に基づいた音素単位の継続長を埋め込んだ時間的特徴量を用いることで, 音素継続時間長が似ている話者同士では, 近い空間に分布される埋め込みベクトルを生成できるようになった．

3 提案手法

本稿では, 韻律情報を明に扱ったマルチファクターな音声仮名化を提案する．提案手法の概要を図 2 に示す．

3.1 F0・発話内容・声色埋め込みの取得

提案手法では, 従来の声色に関する話者特徴量 x-vector の置換に加え, 入力音声から韻律情報の埋め込みを取得, 変換することにより時間的特徴に表れる個人性を明示的に仮名化する．

はじめに, 入力音声 $\boldsymbol{X}$ から基本周波数 (fundamental frequency: F0), 発話内容, 声色に関する話者特徴量 (x-vector) を抽出する．抽出した発話内容から得られる韻律情報と声色に関する話者特徴量に対し, 仮名化のための変換を行う．

3.2 声色情報の仮名化

声色情報における仮名化処理では, 事前訓練済みのベクトルを用いて仮名化を行う．入力音声 $\boldsymbol{X}$ から抽出した x-vector $\boldsymbol{s}$ は, [2] 同様に仮名化処理を施す．入力音声から抽出した $\boldsymbol{s}$ に対し, 事前訓練済み x-vector のプールから複数の x-vector を選択・平均して生成された x-vector $\tilde{\boldsymbol{s}}$ を疑似話者とみなし, 置換する．

3.3 韻律情報の仮名化

同様に, 韻律情報における仮名化では, [8] で提案されている時間的情報を埋め込んだ韻律埋め込みベクトルを用いて仮名化する．韻律情報埋め込み取得の流れを, 図 3 に示す．

まず, 韻律情報の埋め込みプールを用意する．訓練用音声から音声認識モデルを用いて抽出した発話内容 $\boldsymbol{X}_{\text{train}}$ に対し, 音素単位のアライメント $\boldsymbol{P}_{\text{train}}$ を取得する．アライメント $\boldsymbol{P}_{\text{train}} = [p_1, \ldots, p_T]$ は発話内容を各単語の読み (かな), そして音素に変換した形で取得する．ここで, $p_t \in \mathbb{R}^{K+1}$ とし, K は音素の種類とする．アライメント取得により, 発話内の各音素とその継続長を得られる．取得したアライメント $\boldsymbol{P}_{\text{train}}$ は,

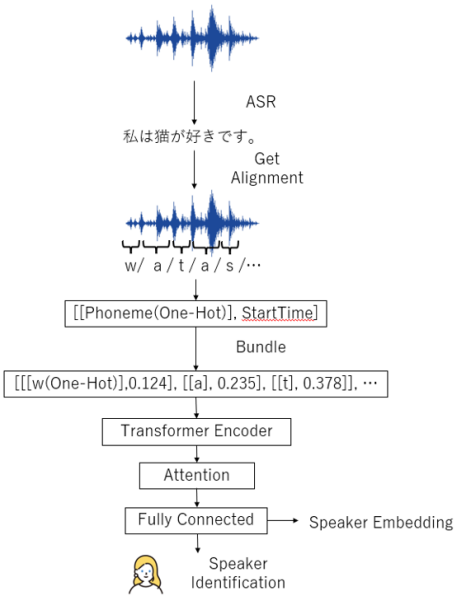

Fig.3: Overview of Embedding Speaker Rhythm Features

各音素の発話内での開始時刻とともに 1 次元のベクトルの形で, Transformer の Encoder に入力する．

$$\boldsymbol{H} = \text{TransformerEncoder}(\boldsymbol{P}_{\text{train}};\ \theta) \tag{1}$$

θ はモデルパラメータ, $\boldsymbol{H}$ は出力シーケンスである．この Transformer を用いたモデルは, 話者認識モデルとして学習する．ここで, [8] でも指摘されているように, 韻律の個人性は発話全体に現れるものであるため, 長期にわたる特徴を学習できる Transformer を採用した．入力音声に対し, Transformer を用いて話者認識ができるよう訓練したモデルの分類層の直前, すなわち全結合層から, 韻律情報の埋込として $\boldsymbol{P}_{\text{train\_emb.}}$ を取得し, プールする．

つづいて, 仮名化対象である音声について, 訓練用音声に対する処理と同様に, 入力音声に対してアライメントを取得し, 韻律埋め込みを加工する．仮名化のため, $\boldsymbol{X}_{\text{train}}$ と同様, 仮名化対象である入力音声から認識した発話内容 $\boldsymbol{X}$ に対し, アライメント $\boldsymbol{P}$ を取得する．取得した $\boldsymbol{P}$ を話者認識タスク用に訓練された Transformer モデルに入力することで, 仮名化対象の音声から全結合層の出力, すなわち入力音声の韻律埋め込み $\boldsymbol{P}_{\text{emb.}}$ を得る．そして, $\boldsymbol{P}_{\text{train\_emb.}}$ の一つ $\tilde{\boldsymbol{P}}_{\text{train}}$ を $\boldsymbol{P}_{\text{emb.}}$ と置換する．

3.4 仮名化済み音声合成

最後に, 取得した発話内容 $\boldsymbol{X}$ と置換後の $\tilde{\boldsymbol{s}}$, $\tilde{\boldsymbol{P}}_{\text{train}}$ を合成する．音声合成で生成されたメルスペクトログラムを Vocoder に渡し, 仮名化済み音声波形を取得する．

4 実験条件

4.1 データセット

本実験では, 入力に CommonVoice 14.0 日本語データセットを用いる．CommonVoice データセットは Mozilla によって公開されたオープンソースデータである．本実験では,

Table 1: Structure of the speech rhythm embedding model implemented using Pytorch. The prosodic information embedding is $(271, 512)$, and 271 is the type of phoneme confirmed at the time of alignment acquisition.

Layer	Output Shape	Configuration
Encoder Input	512	Phoneme Phoneme Duration
MultiHeadAttention	512	NonDynamically QuantizableLinear
Linear1	2048	
Linear2	512	
LayerNorm × 2		eps:1e-5 elementwise_affine:True
Dropout × 2		probability:0.2 inplace:False

CommonVoice14.0 Train データ計 7090 の発話に対し, アライメントを取得できた 6349 発話を用いた. その他のデータは, 元々のデータセットで誤ったスクリプトが付与されているなどの原因からアライメントを取得できなかったため, これらの発話は本実験では使用しない. アライメントの単位は, 日本語話し言葉コーパス (Corpus of Spontaneous Japanese : CSJ) で採用された音韻計 148 個に加え, アライメント取得時に確認したアルファベット等を含む音素計 271 個とする. Transformer で学習する際は, 音素とその開始時間を含む 2 次元ベクトルを, バンドルとして対象の音素を挟む前後 1 個ずつの音素をセットにし, Transformer Encoder に入力した. x-vector は, VoicePrivacy Challenge[1] ベースラインシステムで使用されている x-vector 抽出器 (Kaldi [10] を使い VoxCeleb [11] を用いて学習された抽出器) を利用する. また, 置換先の x-vector を生成するため, 同抽出器を用いて VoxCeleb から抽出された x-vector 計 7325 個で構成されたプールを用いる.

4.2 モデル構造

アライメントは, Common Voice 14.0 日本語データセット (Train, Dev, Validation) を用いて日本語音声認識用にファインチューンした wav2vec 2.0 [12] モデル [*1]を用いて取得した. 各話者の発話から取得したアライメントに基づいて話者照合をするモデルは, [8] でも使用されている Transformer [13] を採用し, Transformer Encoder を利用した. この韻律特徴抽出モデルには, クロスエントロピー誤差を使用した. 学習時のバッチサイズは 64, 実験は 100Epoch 回した. 音声合成は, Tacotron2 [14] を用いてメルスペクトログラムを生成し, HiFi-GAN [15] を用いて新たな音声を合成した. Tacotron2 は, TTS(Text To Speech) アルゴリズムの一つであり, 入力されたテキストをメルスペクトログラムへ変換する. HiFi-GAN は, 音声波形を生成する Generator および 2 つの Discriminator により構成されたニューラルボコーダである.

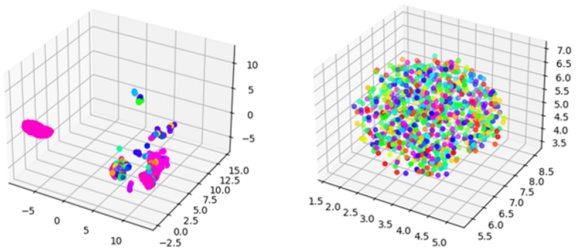

Fig.4: Distribution maps of x-vector(Left) & phoneme duration embedding(Right) compressed and visualized with UMAP.

VoicePrivacy Challenge Baseline System および各モデルの学習環境は, NVIDIA GeForce RTX 3070× 2, メモリ 16GB で実行した.

4.3 評価手法

評価では, VoicePrivacy Challenge の評価基準に準拠し, 生成された仮名化済み音声の仮名化性能, および仮名化後音声における発話内容の保持度合を評価する. 仮名化性能は, VoicePrivacy Challenge で使用されている x-vector, PLDA [16] に基づいた話者照合 (Automatic Speaker Verification: ASV) モデルと, 発話から埋め込んだ話者の韻律情報 [8] に基づいて話者照合を行うモデルの計 2 種を利用する. ASV モデルの等価誤り率 (Equal Error Rate: EER) が高いほど発話者が特定できない, すなわち仮名化性能が高いと評価する.

音声認識には, アライメント取得時に利用した Common Voice 14.0 日本語データセット (Train, Dev, Validation) を用いて wav2vec 2.0 [12] をファインチューンした日本語用音声認識モデル (Automatic Speaker Recognition: ASR) を用いる. ASR モデルの単語誤り率 (Word Error Rate: WER) が低いほど, 発話内容が正しく取得できる, すなわち仮名化済み音声は元の音声の発話内容を高性能に保持していると評価する.

5 実験

表 2 に, CommonVoice 14.0 Train データセットのアライメント取得結果の例を示す. 本稿で提案した手法を検証するにあたり, VoicePrivacy Challenge Baseline System1 を評価した. VoicePrivacy Challenge Baseline System1 は, [2] で提案されている x-vector を用いた音声匿名化手法を実装したものである. ここでは, HiFi-GAN [15] を用いて生成された匿名化済み音声に対する性能を表 3 に示す. 評価には, Libri データセットおよび VCTK コーパス [*2]を用いた. 両データセット共に EER は 10% 前後である一方, VCTK コーパスにおいては Libri データセットよりも匿名化後の音声明瞭度が落ちることが分かる.

また, 図 4 に, CommonVoice 14.0 Train データセットの内,

[*1] huggingface.co/pinot/wav2vec2-xls-r-300m-ja-syllable-cv-14

[*2] https://doi.org/10.7488/ds/2645

Table 2: Example of getting an alignment. ”sentence” is the correct string of input speech, ”phoneme” is the phoneme to be aligned, ”start” is the start time of the target phoneme, and ”end” is the end time of the target phoneme.The unit of time is seconds. The unit of time is seconds.

sentence	phoneme	start	end
日本へ戻ってから, それぞれ, 出世をしている様子であった.	ニ	0.499	0.599
日本へ戻ってから, それぞれ, 出世をしている様子であった.	ッ	0.559	0.999
この肉がいちばん安いです.	コ	0.979	1.059
イさんはどっちの飲み物がいいですか.	イ	0.560	0.700

Table 3: Results of Baseline System 1

		EER ↑	WER ↓
Libri	dev	12.31	4.19
	test	8.64	4.43
VCTK	dev	8.17	10.98
	test	10.49	10.69

Fig.5: Learning curve of the Speech Rhythm-Based Embedding extraction model. The lower the Loss, the more correctly the learning has been done. At first glance, this learning seems to be progressing well, but the output size is $(batch\_size, num\_of\_speakers, num\_of\_speakers)$, making obtaining speaker embeddings with appropriate prosodic information impossible.

韻律情報埋込抽出のために学習した Transformer モデルの訓練で使用した話者 116 名分の x-vector の分布および音素の継続長の埋め込み分布を示す. 可視化は, 512 次元の各 x-vector ならびに PyTorch の EMBEDDING [*3]を用いて音素継続長を埋め込んだベクトルを UMAP [17] を用いて次元削減し, 3 次元空間上にプロットした. この x-vector 群は, JTubeSpeech コーパス [18] の話者 1233 人分のデータを用いて学習された x-vector 抽出器 [*4]を使用して, 各入力音声から抽出した.

図 5 は, 韻律情報特徴量を抽出するモデルの学習曲線である. 損失 (Loss) の変動を示しており, 数値が小さいほど正しく学習しているといえる.

6 考察

本稿では, VoicePrivacy Challenge Baseline System1 の評価を日本語データセットを用いて行い, 韻律情報埋込の抽出モデルも同様に日本語データセットを用いて学習した. 現時点では, 韻律情報抽出モデルが正しく学習できていない. これには, 学習データセットが原因の一つである. まず, 学習データセットについて, [8] では 920 人の話者によるデータを使用しているのに対し, 本研究では 116 人のデータセットで学習をしている. そのため, 100Epoch の学習では十分に 116 人の韻律情報埋込を適切な箇所に分布できず, 匿名化に使用できる性能の話者埋め込み分布は得られない. そして, モデルのネット構造について, 現在は Transformer を用いて学習しており, 入力音声データは最大長のものに合わせゼロパディングを施している. これに関して, RNN を基にした話者認識モデルの内, Encoder 部分を Transformer に置き換えることで, 可変長の入力音声データの韻律情報埋込を学習できるようになるといえる. また, VoicePrivacy Challenge Baseline System1 に対し日本語コーパスを入力した際の性能を分析するため, 本稿では x-vector の分布を可視化した. 図 4 に示すように, x-vector の分布は偏ったものとなっている. ここから, Baseline System1 で提案されている x-vector の匿名化手法 (入力音声から抽出した x-vector に対し, 最も離れた x-vector 100 個を pool から抽出, 平均して疑似話者 x-vector として置換) では, 匿名化後の疑似話者数は限られたものになり, 匿名化の結果によっては, 他の音声と区別がつかなくなることが想定される. また音素継続長の分布から, PyTorch の EMBEDDING のみでは個人性を捉えられず, 原著論文 [8] のような高精度の特徴量を得るには, 話者認識モデルを利用する必要があるといえる. さらに, 現在の声色および韻律における個人性の変換は, ランダムに行うことで個人同定を防いでいる. しかし, 権限のある人が元の発話者を知りたい場合, 提案手法で生成された音声は話者に関する情報が加工されてしまっているため, 復元できない. 元の話者の音声として音声データを復元する場合には, 元の話者の声色および韻律埋め込み情報が必要となり, 復元前に仮名化済み対象音声の話者を把握しなければならない. これは, 元の発話者が誰か知りたいという前提条件と矛盾しており, 音声の復元が可能なケース・目的は限られるといえる.

[*3] `https://pytorch.org/docs/stable/generated/torch.nn.Embedding.html`
[*4] https://github.com/sarulab-speech/xvector_jtubespeech

7 おわりに

本稿では, 従来使用されてきた x-vector を用いた話者特徴量の変換に加え, 話者の韻律情報も変換するマルチファクターな音声仮名化を紹介した. 今後の課題として, 韻律情報の観点に着目した音声仮名化評価法に関する議論と, 利用目的に応じたレベル別の仮名化が挙げられる. その他, 現時点では日本語コーパスを用いた検証を進めているが, 英語や他言語に対してアライメントを取得し韻律情報埋込を抽出, 変換によって音声を仮名化した場合に性能が変化するかどうか検証する.

現在, Common Voice 14.0 日本語用データセットから wav2vec 2.0 を日本語音声認識タスク向けにファインチューンしたモデルを利用し, アライメントを取得した. 取得したアライメントを用いて音素単位前後一つずつを組み合わせたバンドルを生成し, Transformer Encoder を用いて話者認識モデルを学習している. 提案手法実現のため, この話者認識モデルの性能を向上させ, [8] らと同等の韻律情報埋込を抽出する必要がある. CommonVoice 14.0 Train データセットに含まれる話者数は 116 人であるのに対し, VoxCeleb は 1000 人以上とデータセット内に含まれる話者数が大きく異なっており, 今後日本語の韻律情報埋込抽出モデルを学習する際には, データ拡張が不可欠である. 特に, 声色であれば言語ではなく話者に依存するため, 話者の話す言語に関係なく話者数を増やして学習すればよいが, 韻律情報は言語によって異なるため日本語話者数を増やす必要がある. CommonVoice 16.0 には, 8 発話以上揃っている話者が 1252 人, 1 発話以上揃っている話者が合計 3117 人含まれており, 1 発話しかない話者のデータも訓練用データとして使用することで, 現在より幅広い韻律情報の埋め込み分布を取得できると推測する. 日本語話し言葉コーパス (Corpus of Spontaneous Japanese : CSJ) や JTubeSpeech [18] といったコーパスが公開されており, 自由発話を含むこれらのコーパスを利用することで, より個人性が顕著になった韻律情報埋込の分布を取得できるといえる. その他, 現在手元にある CommonVoice データセットに対し, 音声を伸縮させることで話速のパターンを増やすという方法がある.

参考文献

[1] Natalia Tomashenko, et al. The VoicePrivacy 2020 challenge: Results and findings. *Computer Speech &amp Language*, Vol. 74, p. 101362, jul 2022.

[2] Fuming Fang, et al. Speaker anonymization using x-vector and neural waveform models, 2019.

[3] Jose Patino, et al. Speaker anonymisation using the mcadams coefficient, 2021.

[4] David Snyder, et al. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5329–5333, 2018.

[5] S. Pavankumar Dubagunta, et al. Adjustable deterministic pseudonymization of speech. *Computer Speech Language*, Vol. 72, p. 101284, 2022.

[6] Stephen Mcadams. Spectral fusion, spectral parsing and the formation of auditory images. 01 1984.

[7] Carmen Magariños, et al. Reversible speaker de-identification using pre-trained transformation functions. *Computer Speech Language*, Vol. 46, pp. 36–52, 2017.

[8] Kenichi Fujita, et al. Phoneme Duration Modeling Using Speech Rhythm-Based Speaker Embeddings for Multi-Speaker Speech Synthesis. In *Proc. Interspeech 2021*, pp. 3141–3145, 2021.

[9] Kenichi Fujita, et al. Speech rhythm-based speaker embeddings extraction from phonemes and phoneme duration for multi-speaker speech synthesis. *IEICE Transactions on Information and Systems*, Vol. E107.D, No. 1, pp. 93–104, 2024.

[10] Daniel Povey, et al. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. IEEE Catalog No.: CFP11SRW-USB.

[11] Arsha Nagrani, et al. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech Language*, Vol. 60, p. 101027, 2020.

[12] Alexei Baevski, et al. wav2vec 2.0: A framework for self-supervised learning of speech representations, 2020.

[13] Ashish Vaswani, et al. Attention is all you need, 2023.

[14] Jonathan Shen, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions, 2018.

[15] Jungil Kong, et al. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis, 2020.

[16] Simon J.D. Prince, et al. Probabilistic linear discriminant analysis for inferences about identity. In *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, 2007.

[17] Leland McInnes, et al. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

[18] Shinnosuke Takamichi, et al. Jtubespeech: corpus of japanese speech collected from youtube for speech recognition and speaker verification, 2021.