吃音者音声合成のための連発・難発を考慮した合成モデル学習 *

◎高島遼一 [1,2]，△安井美鈴 [3]，滝口哲也 [1]
[1] 神戸大学，[2]JST さきがけ，[3] 大阪人間科学大学

1 はじめに

吃音とは，言葉が流暢に話せない，スムーズに言葉が出てこない発話障害の一種である。本研究ではコミュニケーションの支援を目的として，吃音者の声質でかつ流暢な音声を合成可能なテキスト音声合成（text-to-speech: TTS）システムを検討する。

本研究で収録した吃音者音声のスペクトログラムの一例を Fig. 1 に示す。吃音の症状は，言葉の一部を繰り返して発声する「連発」（こ、こ、こんにちは），言葉の一部を引き伸ばして発声する「伸発」（こーーんにちは），発声に詰まる「難発」（...、こんにちは）の 3 種類に大別される。図に示される音声は「ついで、財務省が、専門家を集めて、具体案を練った。」という台本を読み上げたものであるが,「ついで」の発声中に伸発が,「集めて」の発声時に難発および「あ」の連発が,「具体案」の発声中に難発が生じている。またこの吃音当事者の発話の特徴として，分節ごとにポーズが挿入される（“案を ＜ ポーズ ＞ 練った”）傾向が見られている。台本のテキストを入力，この音声を教師として TTS モデルを学習した場合，入力テキストと実際の発話内容との間でミスマッチが生じるため，生成される音声も非流暢なものとなる。

学習データのテキストと実際の発話とのミスマッチを解消する最も単純な方法は，実際の発話をもとにテキストを手動で修正することである。先行研究 [1] では，テキストを手動で修正して TTS モデルを学習することで，合成音声の流暢さが向上することを確認している。しかしテキストの手動修正はコストが高いため，システムの運用・展開の面で課題が残る。

テキストの修正を自動化する方法として，吃音音声認識モデル [2] を事前に学習しておき，認識結果をもとにテキストを修正する方法が考えられる。しかしこの方法では，吃音音声認識モデルを学習するために，吃音音声と吃音イベントまで詳細に書き起こしたテキストが別途必要である。本研究では，3 種類の吃音症状のうち連発および難発に焦点を当て，吃音音声の書き起こしを使用せずに，テキストを自動で修正しながら TTS モデルを学習する手法を提案する。

2 VITS

高品質な TTS モデルは複数提案されているが [3]，本研究はコミュニケーション支援を目的としているため，リアルタイムで推論が可能な VITS [4] をベースラインとする。本節では，提案手法に関連する部分に絞って VITS の学習について概説する。

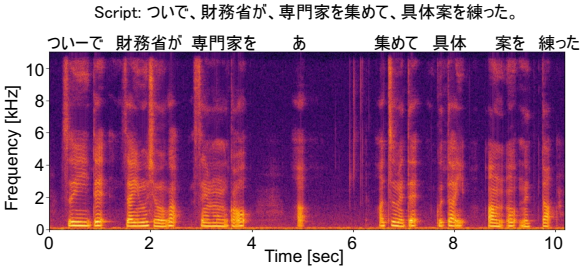


Fig. 1: A spectrogram of the speech recorded from a person who stutter.

VITS の学習の流れを Fig. 2a に示す。VITS の学習は，スペクトログラムから波形を生成する部分（Fig. 2a 左側）と，音声の潜在表現とテキストの潜在表現の対応づけを行う部分（Fig. 2a 右側）に大きく分けられる。左側の処理では，学習データのスペクトログラムから Posterior Encoder によって音声の潜在表現 z を得て，Decoder によって音声波形を再構成する枠組みとなっている。

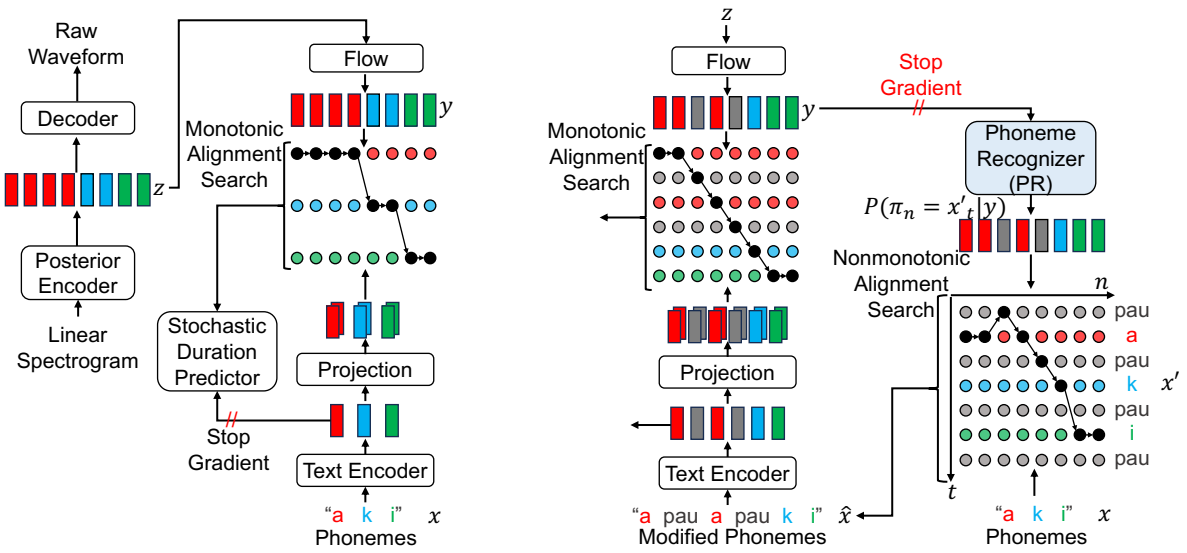
Fig. 2a 右側の処理では，まず音声の潜在表現 z から，Flow と呼ばれるモジュールによって話者性を取り除き，コンテキストに近い表現 y を得る。一方学習データのテキスト（音素列）から Text Encoder と Projection モジュールによってテキストの潜在表現を得る。ここで，音声の潜在表現とテキストの潜在表現それぞれの確率分布を近づけるための損失関数を計算するが，そのためには両者の時間的対応（アライメント）を取る必要がある。そのため，Monotonic Alignment Search (MAS) と呼ばれる Viterbi アルゴリズム [5] をベースとした手法により，最適なアライメントパスを推定する。

3 提案手法

吃音音声を用いて VITS を学習する際の問題点は，例えば台本 “a k i” を吃音当事者が読み上げた音声が，実際は “a pau a pau k i”(“pau” はポーズを表す) という発話になっていた場合，ポーズや連発を含む音声区間が MAS によって単一の音素として対応付けられてしまうことである。従って，ポーズと連発の存在をテキストに反映させ，ポーズはポーズ，連発は連発として正しく対応させる必要がある。

提案するテキストの自動修正方法を Fig. 2b に示す。まず，Flow モジュールの出力系列 y から，各音声フレームに対する音素の事後確率を計算する Phoneme Recognizer (PR) を新たに導入する。PR は 3 層の双

*Training of speech synthesis model considering repetitions and blocks for stutterer-oriented text-to-speech. by TAKASHIMA, Ryoichi[1,2], YASUI, Misuzu[3] TAKIGUCHI, Tetsuya[1] ([1]Kobe University, [2]JST PRESTO, [3]Osaka University of Human Sciences)



(a) Original VITS (b) VITS with pause and repetition detector

Fig. 2: Training procedures of (a) original VITS and (b) proposed method.

方向 GRU で構成されており，softmax 関数により 38 種類の音素にポーズトークン “pau” を足した 39 トークンの事後確率をフレームごとに計算する。この事後確率を用いて，ポーズおよび連発の挿入を考慮したアライメントパスを求める。

仮に台本テキストのトークン列を $x = \{\mathrm{a}, \mathrm{k}, \mathrm{i}\}$ としたとき，ポーズの挿入を考慮するため，音素列の先頭，末尾，および各音素の間にポーズトークンを挿入した，変形トークン列 $x' = \{\mathrm{pau}, \mathrm{a}, \mathrm{pau}, \mathrm{k}, \mathrm{pau}, \mathrm{i}, \mathrm{pau}\}$ を作成し，トークン列 x の代わりに x' を用いてアライメントを推定する。音声フレームごとに対応する音素を並べたアライメント系列を π としたとき，π 以下の式によって推定される。

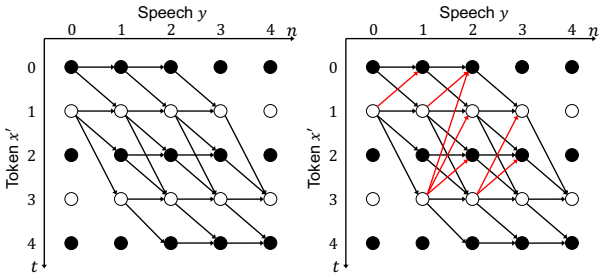
$$\hat{\pi} = \arg\max_{\pi} \sum_{n} \log P(\pi_n | y). \tag{1}$$

ここで，$P(\pi_n|y)$ は PR によって計算される，フレーム n におけるトークンの事後確率分布である。

確率最大のアライメントパス $\hat{\pi}$ は，Viterbi アルゴリズムをベースとした手法により求められる。このとき，ポーズや連発を任意に挿入可能にするため，遷移パスに複数の制約を設けている。まず，ポーズの挿入のみを考慮した場合に許可される遷移パスを Fig. 3a に示す。ポーズを挿入した変形トークン列 $x' = \{x'_0, \ldots, x'_t, \ldots, x'_{T-1}\}$ について，インデクスが奇数のトークン（図中の白丸）は台本テキスト x に元々存在する音素で，インデクスが偶数のトークン（図中の黒丸）は挿入されたポーズトークンである。

音声フレーム $n = 0$ のとき，あり得るトークンは x'_0（文頭にポーズが挿入）および x'_1 のみである。そこで，初期状態を以下のように定義する。

$$Q_{t,n=0} = \begin{cases} \log P(\pi_0 = x'_t | y) & (\text{if } t = 0 \text{ or } 1) \\ -\inf & (\text{otherwise}) \end{cases} \tag{2}$$



(a) Pause only (b) Pause and repetition

Fig. 3: Possible alignment paths considering (a) only pause and (b) pause and repetition.

$Q_{t,n}$ はフレーム n $(0 \le n \le N-1)$，トークンインデクス t $(0 \le n \le T-1)$ の点に到達するまでに計算される対数確率の最大値である。

次に，音声フレーム $0 < n \le N-1$ の場合について考える。このとき遷移パスとして，「音素から次の音素に遷移する」，「ポーズ/音素から音素/ポーズに遷移する」，「同じトークンに留まる」というパターンのみを考慮する。よって，$0 < n \le N-1$ における $Q_{t,n}$ は以下の漸化式によって定義される。

$$Q_{t,n} = \log P(\pi_n = x'_t | y) + \begin{cases} \max(Q_{t-2,n-1}, Q_{t-1,n-1}, Q_{t,n-1}), & (\text{if } t \text{ is odd}) \\ \max(Q_{t-1,n-1}, Q_{t,n-1}) + \log w_{\mathrm{p}} & (\text{if } t \text{ is even}) \end{cases} \tag{3}$$

ここで，w_{p} $(0 < w_{\mathrm{p}} \le 1)$ はポーズ挿入に対するペナルティであり，$w_{\mathrm{p}} = 0.9$ としている。

遷移の最終状態は，文末にポーズが挿入されるか否かの 2 パターンであるため，最適なアライメント

パスの対数事後確率は以下の通りである。

$$\max_{\pi} \sum_{n} \log P(\pi_n|y) = \max(Q_{T-2,N-1}, Q_{T-1,N-1}) \quad (4)$$

次に，ポーズに加えて連発も考慮した場合に許可される遷移パスを Fig. 3b に示す。連発は発声しようとしている語の一部を繰り返す現象であるため，図中の赤線で示されるような，t のマイナス方向への遷移で連発を表現する。このとき，考慮する連発に含まれる音素数を K ($K > 0$) とする。例えば「お、お、おはよう」の場合は $K = 1$（“お: o”）,「こん、こんにちは」の場合は $K = 3$（“こん: k o N”）である。Fig. 3b は $K = 2$ とした場合の遷移パスを図示している。考慮する連発の音素数 K に対して，t のマイナス方向へ遷移する最大幅 K' は $K' = 2K - 1$ となる。ただし，本手法において，マイナス方向への遷移は音素（図中の白丸）からのみ許可しており，ポーズ（図中の黒丸）からの遷移は許可しない。このとき，$Q_{t,n}$ の漸化式は以下のようになる。

$$Q_{t,n} = \log P(\pi_n = x'_t|y) + \begin{cases} \max(Q_{t-2,n-1}, Q_{t-1,n-1}, Q_{t,n-1}, \\ \quad Q_{t+2,n-1} + \log w_{\mathrm{r}}, \\ \quad Q_{t+4,n-1} + \log w_{\mathrm{r}}, \ldots, \\ \quad Q_{t+K'-1,n-1} + \log w_{\mathrm{r}}), & (\text{if } t \text{ is odd}) \\ \max(Q_{t-1,n-1}, Q_{t,n-1}, \\ \quad Q_{t+1,n-1} + \log w_{\mathrm{r}}, \\ \quad Q_{t+3,n-1} + \log w_{\mathrm{r}}, \ldots, \\ \quad Q_{t+K',n-1} + \log w_{\mathrm{r}}) & (\text{if } t \text{ is even}) \end{cases} \quad (5)$$

ここで，w_{r} ($0 < w_{\mathrm{r}} \leq 1$) は連発挿入に対するペナルティであり，$w_{\mathrm{r}} = 0.8$ としている。

以上の方法により，確率が最大となるアライメントパス $\hat{\pi}$ を求めることでポーズおよび連発を検知し，それらを反映したテキスト $\hat{x}$ が得られる。以降，台本テキスト x の代わりに修正テキスト $\hat{x}$ を用いて，従来の VITS と同様の学習処理を行う。

PR は VITS の他のモデルパラメータと同時に学習される。PR を学習するための損失関数は，得られた最適アライメントパス $\hat{\pi}$ を用いて，以下の重み付きクロスエントロピーによって定義される。

$$L = -\sum_{n} w(\hat{\pi}_n) \log P(\hat{\pi}_n|y) \quad (6)$$

ここで，重み $w(\hat{\pi}_n)$ は「台本テキスト x 中に π_n と同じ音素が含まれる数 / アライメント $\hat{\pi}$ 中に $\hat{\pi}_n$ と同じ音素が含まれる数」で計算される。単にアライメントパスとのクロスエントロピー損失を使用した場合，例えば母音のような多くのフレームに割り当てられる音素に対する損失が相対的に大きくなるため，子音のような少ないフレームに割り当てられる音素の学習が進みにくくなる。そのため，各音素について，アライメント中の出現頻度で正規化することで，PR の学習を効率化している。また，Flow 出力 y と PR の間は Stop Gradient を行っており，損失 L が他のモジュールに影響しないようにしている。これは，PR の導入が VITS の合成性能に影響を与えないようにするためである。

Table 1: Phoneme error rates (PER) [%] of the original script and that modified by the proposed method. “pau” indicates the PER associated with pause token.

	pau	Total
Original script	4.39	6.32
+ Pause insertion ($K = 0$)	1.98	3.91
+ Repetition insertion ($K = 2$)	1.52	3.69
+ Repetition insertion ($K = 4$)	1.10	3.01
+ Repetition insertion ($K = 6$)	1.19	3.53
+ Repetition insertion ($K = 8$)	1.53	3.74

4 実験

4.1 実験条件

吃音を持つ女性話者 1 名が ATR 日本語音声データベースの音素バランス文 503 文を読み上げた音声を収録した。音声のサンプリング周波数は 22,050 Hz である。収録音声 503 発話のうち 50 発話は評価に使用し，残りの 453 発話のうち，発話長が一定以上のものやクリッピングが生じている音声を除いた 363 発話を学習に用いた。

提案手法は VITS の公式プログラム [4] に対して実装した。JSUT コーパス [6] 用いて VITS を事前学習し，その後吃音音声を用いてファインチューニングを行った。JSUT コーパスで事前学習する際は，ポーズおよび連発の挿入は起こらないように，つまり $w_p = w_r \fallingdotseq 0$，$K = 0$ として学習を行った。入力トークン列は ESPNet2 [7] の JSUT データセット用レシピで用意されている “Phoneme+Accent+Pause” の設定を参考に，音素と二種類のアクセント記号を用いて定義した。ただし提案手法のテキスト修正用アライメント推定の際はアクセント記号は除去し，音素列のみを用いた。

4.2 実験結果

まず，提案手法によるテキスト修正の精度を評価するため，手動で修正したテキストをリファレンスとして，オリジナルの台本テキストと，提案手法により自動修正したテキストの音素誤り率（Phoneme error rate: PER）を計算した。結果を Table 1 に示す。表中の “Total” の列は全体の PER を示し，“pau” の列はポーズトークンに関連する誤りのみをカウントした誤り率を示す。連発を考慮せず（$K = 0$）にポーズ挿入のみを行なった修正テキストは，ポーズに関連する誤り率が 2.41%低下しており，同じ分だけ全体の誤

Table 2: MCDs [dB] from the synthesized speech when the model was trained with manually modified text and the average durations (dur) [sec] per utterance.

	MCD	dur
Original script	5.43	6.47
+ Pause insertion ($K = 0$)	4.84	6.02
+ Repetition insertion ($K = 4$)	4.69	5.74
Manually modified text	-	5.68

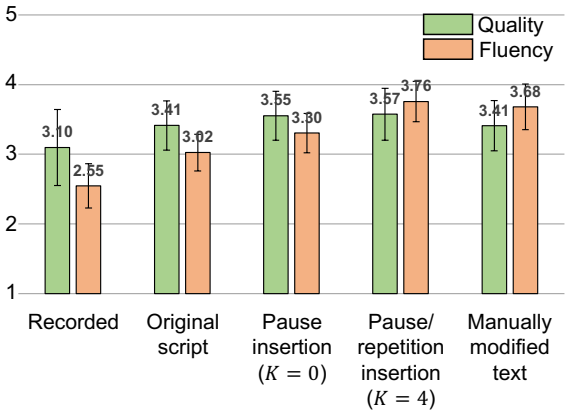


Fig. 4: Mean opinion scores for speech quality and fluency. Error bars represent the 95 % confidence intervals.

り率も低下した。また，連発の挿入も行なった修正テキストはさらに誤り率が低下しており，考慮する連発の音素数 $K = 4$ の時が最も低い誤り率を示した。

次に，オリジナルの台本テキストを学習に使用した場合と，提案手法により修正したテキストを使用した場合，そして手動で修正したテキストを使用した場合で，それぞれの合成音声を評価した。客観評価として，手動修正したテキストを使用した場合の合成音声をリファレンスとして計算したメルケプストラム歪み（Mel-cepstral distortion: MCD），および 1 発話あたりの平均発話長を評価した。

客観評価結果を Table 2 に示す。Table 1 において低い PER を示した条件ほど MCD が小さくなっていることから，手動修正テキストに近いテキストを用いることで，合成される音声も手動修正テキストを使用した場合と近くなることがわかる。また発話の平均継続長も，手動修正テキストに近いテキストを用いたものほど短くなっている。これは，テキストと実際の発話とのミスマッチを解消することで，合成音声中に含まれる意図しないポーズ挿入が減ったためである。

最後に，各条件で合成した音声について品質と流暢さに関する主観評価実験を行なった。15 名の日本人聴者が，収録音声および各条件での評価用合成音声をそれぞれ 20 発話ずつ，ヘッドフォンを用いて聴いて評価した。品質については音声の品質を，流暢さについては話速の自然さや発声が詰まらずスムーズに発話されているかを，それぞれ 5 段階で評価した。結果を Fig. 4 に示す。まず品質については，いずれの手法も有意差は見られなかった (t 検定，$p > 0.05$)。このことから，提案手法は VITS の合成品質に悪影響を及ぼさないことが示される。しかし，収録音声に背景ノイズや残響が含まれていたことからスコアが低く，またその音声を用いて学習した合成音声の品質も 4 を下回る結果となった。

流暢さについては収録音声が最もスコアが低く，テキストを適切に修正した合成音声ほどスコアが高くなる傾向となった。手動修正テキストを用いた場合と，ポーズと連発を挿入したテキストを用いた場合では有意差はなく，ほぼ同等のスコアを示した。このことから，提案手法によるテキストの自動修正により，手動修正と同等の音声が合成できることが示された。

5 おわりに

本稿では，吃音の症状を考慮して台本テキストを修正しながら音声合成モデルを学習する手法を提案した。今後は合成品質の改善と，VITS 以外の合成モデルへの適用を検討する。

謝辞 本研究の一部は，JST さきがけ JPMJPR23I7 および JSPS 科研費 JP23K20733，JP22K12168 の支援を受けたものである。

参考文献

[1] 長久保諒 他, “吃音者向け TTS システムのための健常者音素継続長を反映した VITS の学習手法の提案,” 音講論（春）, pp. 919–922, 2024.

[2] O. Shonibare et al., “Enhancing ASR for stuttered speech with limited data using detect and pass,” *ArXiv*, vol. abs/2202.05396, 2022.

[3] 岡本拓磨, “ニューラル音声合成技術,” 情報通信研究機構研究報告, vol. 68, no. 2, pp. 47–55, 2022.

[4] J. Kim et al., “Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech,” *ICML*, pp. 5530–5540, 2021.

[5] L.R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[6] R. Sonobe et al., “JSUT corpus: free large-scale japanese speech corpus for end-to-end speech synthesis,” *ArXiv*, vol. abs/1711.00354, 2017.

[7] T. Hayashi et al., “Espnet2-TTS: Extending the edge of TTS research,” *ArXiv*, vol. abs/2110.07840, 2021.