Machine Speech Chainに基づく半教師あり学習を用いた日英コードスイッチング音声の認識

中山 佐保子[1]　チャンドラ アンドロス[1,2]　サクティ サクリアニ[1,2]　中村 哲[1,2]
[1] 奈良先端科学技術大学院大学 情報科学研究科
[2] 理化学研究所 革新知能統合研究センター
{nakayama.sahoko.nq1, andros.tjandra.ai6, ssakti, s-nakamura}@is.naist.jp

1 はじめに

コードスイッチング (CS) とは，同じ会話の中で言語が別の言語に切り替わる現象をいう．CS は入力が多言語になるので，音声認識（ASR）や音声合成（TTS）といった音声言語処理技術で扱うのが難しい．これまでの既存手法では，ASR か TTS のどちらかを CS データで教師あり学習を行うものが多かった．しかし，音声とテキストがパラレルな CS データは入手しにくいため，ラベル付きの CS データを多く必要とする教師あり学習は実用的ではない．そこで，私達は，深層学習のスピーチチェーンを利用して ASR と TTS をループ結合し，日英 CS を半教師あり学習させることにした．まずは，人間が学校で複数の言語を学ぶように，ラベル付きの日本語と英語のモノリンガルデータで ASR と TTS を別々に学習 (教師あり学習) し，その後，人間が CS を聞いて話すように，CS のテキストか音声のどちらかでスピーチチェーンを試した (教師なし学習)．この実験の結果，スピーチチェーンによって ASR と TTS が互いに学習し合い，ラベルなしの CS データで，パフォーマンスが向上することを示した．

2 コードスイッチングの半教師あり学習のためのスピーチチェーン

以前，我々の研究室では，ヒューマンスピーチチェーン [1] を基に，深層学習を用いたマシーンスピーチチェーンを設計した [2, 3]．ヒューマンスピーチチェーンとは，人間にとって大事なコミュニケーションの仕組みで，聴覚フィードバックを行いながらスピーキングとリスニングを繰り返す仕組みである．これに基づくマシーンスピーチチェーンは，コンピュータにスピーキング（ASR）とリスニング（TTS）のどちらかだけではなく，スピーキングとリスニングを同時にできるようにした．Sequence-to-Sequence の ASR [4, 5] と Sequence-to-Sequence の TTS [6]，それらをつなぐループ結合で構成され，そのループ構造が，ラベル付きのデータとラベルなしのデータの両方を組み合わせてモデルを学習できるようにしている．

CS の ASR と TTS は，以下の学習プロセスをもつスピーチチェーンフレームワーク (Fig. 1) で構築される．

1. **ラベル付きのモノリンガルデータで ASR と TTS を別々に学習 (教師あり学習)**
   まず，人間が学校で複数の言語を学ぶのと同じように，ASR と TTS を別々に，ラベル付きのモノリンガルの日本語と英語のデータで学習する（教師あり学習）(図 1(a))．

2. **ラベルなしの CS データを用いてスピーチチェーンで ASR と TTS を同時に学習する（教師なし学習）**
   その後，人間がマルチリンガルの環境で CS を聞いて話すように，ラベルなしの CS データを用いてスピーチチェーンで ASR と TTS を同時に学習する（教師なし学習） (図 1(b)).

   この教師なし学習の学習プロセスを以下の構造に展開する．

   (a) **CS テキストのみが与えられた時の TTS から ASR への展開プロセス**
   CS テキストの入力 y^{CS} が与えられると，TTS は音声波形 $\hat{x}^{CS}$ を作り，作られた音声から ASR がテキスト $\hat{y}^{CS}$ を再構築する（図 1(c)）．そして，出力テキストの予測べ

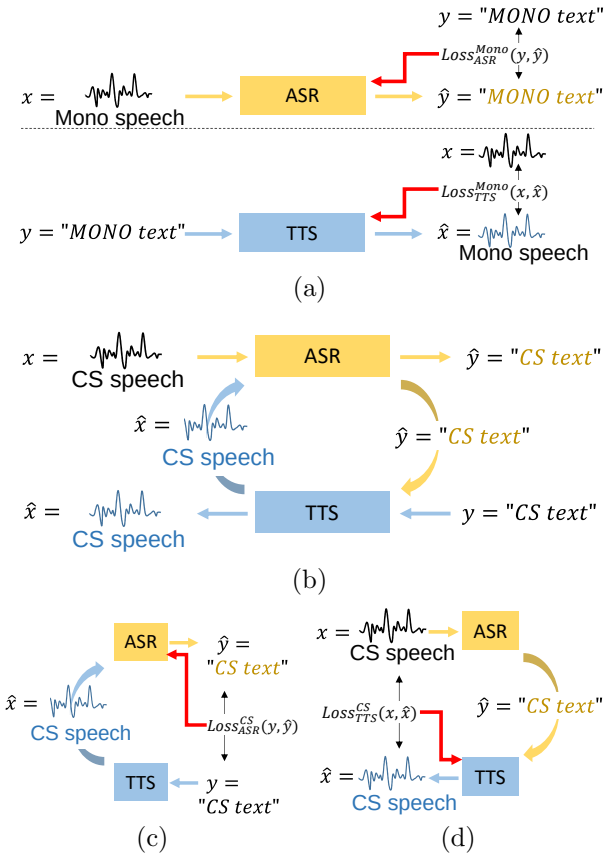

図 1: 提案したフレームワークの概観: (a) ラベル付きのモノリンガルデータで ASR と TTS を別々に学習 (教師あり学習); (b) ラベルなしの CS データで ASR と TTS をスピーチチェーンを用いて同時に学習 (教師なし学習); (c) CS テキストだけが与えられたときの TTS から ASR へのプロセス; (d) CS 音声だけが与えられたときの ASR から TTS へのプロセス.

クトル $\hat{y}^{CS}$ と入力テキスト y^{CS} の間の損失 $L^{CS}_{ASR}(\hat{y}^{CS}, y^{CS})$ を計算し ASR のパラメータを更新する.

(b) **CS 音声のみが与えられた時の ASR から TTS への展開プロセス**
ラベルなしの CS 音響特徴量 x^{CS} が与えられると，ASR はラベルなしの入力音声 $\hat{y}^{CS}$ を認識し，その出力テキストから TTS が音声波形 $\hat{x}^{CS}$ を再構築する（図 1(d) 参照）. そして再構築された音声波形 $\hat{x}^{CS}$ と元の音声波形 x^{CS} の間の損失 $L^{CS}_{TTS}(\hat{x}^{CS}, x^{CS})$ を計算し，TTS のパラメータを更新する.

以下の式で，全ての損失を一つの損失変数に重みづけする.

$$L = \alpha * (L^{Mono}_{ASR} + L^{Mono}_{TTS}) + \beta * (L^{CS}_{ASR} + L^{CS}_{TTS}) \quad (1)$$

$$\theta_{ASR} = Optim(\theta_{ASR}, \nabla_{\theta_{ASR}} L) \quad (2)$$

$$\theta_{TTS} = Optim(\theta_{TTS}, \nabla_{\theta_{TTS}} L), \quad (3)$$

α と β は教師あり（ラベル付き）と教師なし（ラベルなし）の損失を重みづけするハイパーパラメータである.

3 実験

3.1 モノリンガルと CS コーパス

我々は，ATR の旅行会話コーパス（BTEC）[7] の日本語と英語を利用した．ランダムに選んで訓練セットとし，それ以外の 500 文を開発セット，別の 500 文をテストセットとした．日英 CS データセットについては，日本語と英語の BTEC 文から作成した．ここでは，文中 CS の単語レベルとフレーズレベルの 2 種類を作成した．詳細は [8] に記してある.

英語のテキストについては，全ての文字を小文字に変換し句読点記号 [,:?.] を取り除いた．日本語のテキストは，形態素解析 の Mecab[1] を用いてカタカナに変換し pykakasi[2] を用いてアルファベットに変換した．結果の語彙は，26 の文字（a-z）と日本語の音を伸ばすための記号 (-) と文の始めと終わり，単語の間のスペースを表す 3 つのタグ (<s>, </s>, <spc>) となった.

また，テキストデータから音声を作成するのに，Google 音声合成[3] を利用して，モノリンガルの日本語と英語，そして日本語 TTS と英語 TTS の音声を組み合わせて日英 CS の音声を作成した.

この 16kHz サンプリング音声から，Librosa ライブラリ[4] を用いて窓幅 50ms シフト幅 12.5ms で，40 次元の対数メルスペクトログラムと 1025 次元の対数スペクトログラムを抽出し，平均 0 分散 1 に正規化した.

3.2 ASR と TTS システム

用いた ASR はアテンション機構（attention mechanism）を用いたエンコーダデコーダモデル（encoder-decoder model）[4] である．エンコーダは， 各方向に

[1] https://github.com/taku910/mecab
[2] Pykakasi–https://github.com/miurahr/pykakasi
[3] https://pypi.python.org/pypi/gTTS
[4] Librosa–https://librosa.github.io/librosa/0.5.0/index.html

256 の隠れユニットをもつ 3 層の BiLSTM（双方向 512 ユニット）で構成され，活性化関数には LeakyReLU ($l = 1e-2$) [9] を用いた．デコーダは，128 次元の埋め込み層と 512 の隠れユニットをもつ 1 層の LSTM で構成される．

TTS は Sequence-to-Sequence の TTS (Tacotron) [6] をベースにしている．ハイパーパラメータはオリジナルの Tacotron とほぼ同じであるが，ReLU の代わりに LeakyReLU を用いた．また，エンコーダの CBHG では，GPU のメモリ消費を減らすために，1 次元畳み込みバンクは 16 ではなく 8 セットのフィルタを用いた．デコーダでは，256 の隠れユニットをもつ GRU の代わりに 2 層の LSTM を用いている．

ASR と TTS モデルは，どちらも PyTorch ライブラリ[5]で実装した．

4 実験結果

実験結果は次の 3 種類のテストセットで評価した．(1) **TstJa (JaTTS)**: 日本語と対応する日本語 TTS で作られた音声; (2) **TstCS (MixTTS)**: 日英の文中 CS と対応する日本語 TTS と英語 TTS で作られた音声; (3) **TstEn (EnTTS)**:英語と対応する英語 TTS で作られた音声．

ASR の性能は文字誤り率 (CER) で評価し，TTS の性能は真の値と予測した対数メルスペクトログラムの間の L2 ノルムの差を計算した．表 1 に，ベースラインおよび提案する CS スピーチチェーンの ASR および TTS の性能を示してある．

ベースラインはスピーチチェーンを用いずに Sequence-to-SequenceASR および TTS を用いて教師あり学習を行った．ベースラインは，次の 3 種類である．(1) **Ja50k (JaTTS)**: 50k の日本語テキスト，対応する日本語 TTS による音声で学習した ASR および TTS; (2) **En50k (EnTTS)**: 50k の英語のテキスト，対応する英語 TTS による音声で学習した ASR および TTS; (3) **Ja25k+En25k (MixTTS)**: 25k の日本語テキストと 25k の英語テキスト，対応する日本語 TTS と英語 TTS による音声で学習した ASR および TTS.

表 1 を参照すると，Ja50k (JaTTS) ASR は日本語のテストでは良かったが，英語のテストでは非常に悪かった．一方，En50k (EnTTS) ASR は英語テストで非常に低い CER だが，日本語テストでは高い CER だった．TTS も，同じ傾向を示した． Ja25k+En25k (MixTTS) は日本語 TTS で合成した日本語と英語 TTS で合成した英語で学習されたモデルだが，日本語，英語，日英 CS で上手くバランスを取っている．

これに対して，提案システムは，モノリンガルでの性能を維持しながら，ラベル付きの CS データを必要とせずに ASR と TTS がより上手く CS 入力を扱えるようにする．そのために，我々はスピーチチェーンを利用した．

(1) [**ラベル付き Ja25k+En25k (MixTTS)**]+[**ラベルなし CS (JaTTS)**]: ラベル付きデータとして Ja25k+En25k (MixTTS) を用い，ラベルなしデータとしてコードスイッチングの CS (JaTTS) を用いて半教師あり学習した ASR および TTS; (2) [**ラベル付き Ja25k+En25k (MixTTS)**]+[**ラベルなし CS (Mix TTS)**]: ラベル付きデータとして Ja25k+En25k (Mix TTS) を用い，ラベルなしデータとしてコードスイッチング CS (Mix TTS) を用いて半教師あり学習した ASR および TTS; (3) [**ラベル付き Ja25k+En25k (MixTTS)**] +[**ラベルなし CS (Mix+JaTTS)**]: ラベル付きデータとして Ja25k+En25k (MixTTS) を用い，ラベルなしデータとしてコードスイッチングの CS (MixTTS) と CS (Ja TTS) を両方用いて半教師あり学習した ASR および TTS.

我々の提案するスピーチチェーンモデルは，ラベルなしの CS データで ASR と TTS が互いに学習し合い，CS テスト TstCS (MixTTS) で CER18.11% から 5.08%まで (13.03%の減少) 大きく ASR の性能を改善し，モノリンガルの設定で良い性能を維持した (日本語と英語のテストは，それぞれ 0.14% と 1.8%の僅かな減少にとどまっている)．同じ傾向が TTS の結果でも示され，CS テスト TstCS (MixTTS) では L2 ノルムが 0.489 から 0.372 まで性能を改善し，日本語と英語のモノリンガルテストについては性能を維持した．

5 おわりに

日英 CS の ASR のための半教師あり学習によるスピーチチェーンを提案した．まずは，ラベル付きのモノリンガルデータで ASR と TTS を別々に学習 (教師あり学習) し，その後，CS のテキストか音声のどちらかでスピーチチェーンを試した (教師なし学習)．この実験の結果，スピーチチェーンによって ASR と TTS が互いに学習し合い，モノリンガルでの性能を維持しながら，学習にラベル付きの CS データを必要とせずに，CS でのパフォーマンスが向上することを示した．

[5]https://github.com/pytorch/pytorch

表 1: ベースラインと提案する CS スピーチチェーンの ASR の性能（CER）および TTS の性能（L2 ノルム）.

	TstJa(JaTTS)		TstCS(MixTTS)		TstEn(EnTTS)	
	ASR	**TTS**	**ASR**	**TTS**	**ASR**	**TTS**
ベースライン: パラレルな音声テキスト → 教師あり学習						
Ja50k(JaTTS)	2.11%	0.321	33.76%	0.484	81.12%	0.667
En50k(EnTTS)	86.42%	0.373	66.16%	0.469	2.30%	0.417
Ja25k+En25k (MixTTS)	1.71%	0.312	18.11%	0.489	2.99%	0.437
スピーチチェーン: ラベル付き **Ja25k+En25k (MixTTS)** + ラベルなし **CS (JaTTS)** → 半教師あり学習						
+CS10k (JaTTS)	1.85%	0.311	19.66%	0.484	4.79%	0.444
+CS20k (JaTTS)	1.85%	0.306	17.21%	0.489	4.65%	0.441
スピーチチェーン: ラベル付き **Ja25k+En25k (MixTTS)** + ラベルなし **CS (MixTTS)** → 半教師あり学習						
+CS10k (MixTTS)	1.81%	0.312	5.35%	0.374	3.69%	0.437
+CS20k (MixTTS)	1.85%	0.310	5.54%	0.368	3.64%	0.440
スピーチチェーン: ラベル付き **Ja25k+En25k (MixTTS)** + ラベルなし **CS (Mix+JaTTS)** → 半教師あり学習						
+CS20k (Ja+MixTTS)	1.82%	0.305	5.08%	0.372	4.05%	0.439

今回，この研究では，日本語と英語の組み合わせに焦点を当てているが，大きな修正なく他の言語の組み合わせにも応用できると考えられる．また，今後は，複数話者の自然な日英 CS 音声データで実験を行っていく予定である．

6 謝辞

本研究は科研費 JP17H06101，JP17K00237 の助成を受けております．

参考文献

[1] Peter B. Denes and Elliot N. Pinson, *The Speech Chain: The Physics And Biology Of Spoken Language*, Anchor books. Worth Publishers, 1993.

[2] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, “Listening while speaking: Speech chain by deep learning,” in *Proc. of IEEE ASRU*, Okinawa, Japan, 2017, pp. 301–308.

[3] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura, “Machine speech chain with one-shot speaker adaptation,” in *Proc. of INTERSPEECH*, Hyderabad, India, 2018, p. to appear.

[4] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio, “End-to-end attention-based large vocabulary speech recognition,” *CoRR*, 2015.

[5] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *Proc. of ICASSP*, Shanghai, China, 2016, pp. 4960–4964.

[6] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous, “Tacotron: A fully end-to-end text-to-speech synthesis model,” in *Proc. of INTERSPEECH*, Stockholm, Sweden, 2017, pp. 4006–4010.

[7] Genichiro Kikui, Eiichiro Sumita, Toshiyuki Takezawa, and Seiichi Yamamoto, “Creating corpora for speech-to-speech translation,” in *Proc. of EUROSPEECH*, Geneva, Switzerland, 2003, pp. 381–384.

[8] Sahoko Nakayama, Takatomo Kano, Quoc Truong Do, Sakriani Sakti, and Satoshi Nakamura, “Japanese-english code-switching speech data construction,” in *Proc. of Oriental COCOSDA*, Miyazaki, Japan, 2018.

[9] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li, “Empirical evaluation of rectified activations in convolutional network,” *CoRR*, 2015.