【本審査】End-to-endニューラル音声合成のための構文的ピッチアクセントと音素継続長のモデル化

Lexical pitch accent and duration modeling for neural end-to-end text-to-speech synthesis

総研大D3　山岸研究室
安田裕介

博士論文のタイトル

End-to-endニューラル音声合成のための

構文的ピッチアクセントと音素継続長のモデル化

- テキスト音声合成（TTS）
  - テキストを読み上げ音声に変換する技術
- End-to-end音声合成
  - テキスト音声合成手法の1つ。ニューラルネットワークを用いて、単一のモデルのみでテキストから音声への変換を行う。
- 構文的ピッチアクセント
  - 単語や句に紐づくピッチ変化を伴うアクセント
- 音素継続長
  - 音素が音声として継続する長さ

博士論文の構成と発表内容

導入

1. Introduction
   - 1.1. 背景
   - 1.2. スコープと動機

End-to-end TTS とその問題

2. End-to-end text-to-speech synthesis
3. Analysis and issues of end-to-end TTS
   - 3.1. End-to-end TTSの問題1：平坦なピッチ変化
   - 3.2. End-to-end TTSの問題2：致命的アラインメントエラー

ピッチアクセント

4. Learning pitch accent in end-to-end TTS
5. Modeling pitch accent in end-to-end TTS

音素継続長

6. Modeling monotonic alignment in end-to-end TTS
7. Modeling duration in end-to-end TTS

結論

8. Conclusion
   - 8.1. 解決した問題
   - 8.2. 未解決の問題

1. 導入:背景

End-to-end TTSは1つのモデルでテキストから音声に直接変換する手法

☆利点

- 発音を辞書に頼らずテキストから解決するので、言語に対する汎用性の点で期待できる手法
- 伝統的なTTSは1つの機能に特化した複数のモデルから構成されるのに対し、1つのモデルのみ
- 伝統的なTTSでは複雑な言語特徴量ラベルが必要なのに対し、テキストのみでよい

☆課題点

1) 音声のあらゆる特徴をテキストのみから学習するEnd-to-end法では、言語特有の特徴を考慮することが十分になされていない
   a) 研究開始時点では、主に英語を対象に研究されていた
   b) ピッチアクセント言語や声調言語への応用が確立されていなかった
2) 発音の頑健性に課題がある
   a) どのような条件でテキストから発音の学習がうまく働くのか十分に検証されていなかった
   b) アラインメント(テキストと音声の対応関係)が特に不安定で、致命的な発音となる
3) 現在のEnd-to-end法では、モデル内部のコンポーネントがTTSの特徴を考慮してそれに特化した形で設計されていない

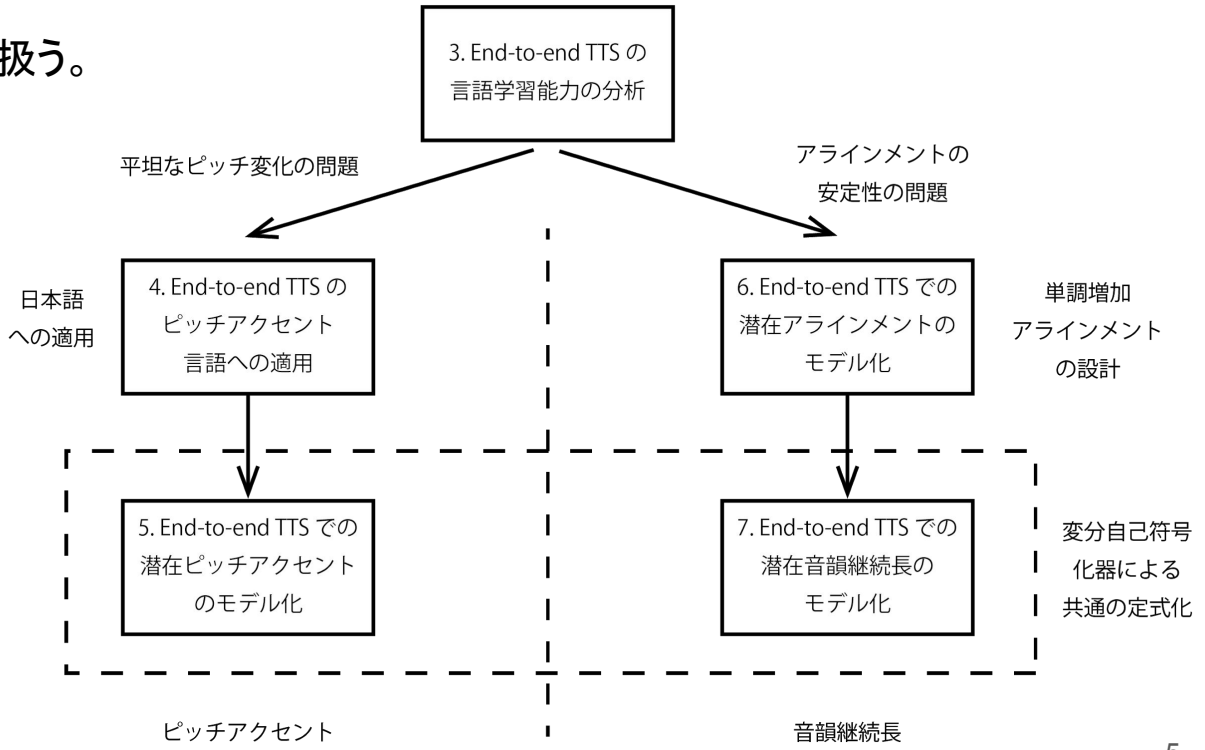
1. 導入：スコープと動機、論文の構成

分析結果（3章）に基づき、
音声における2つの構文的特徴量を扱う。

1) ピッチアクセント
2) 音素継続長

以下の2つを目標とする。

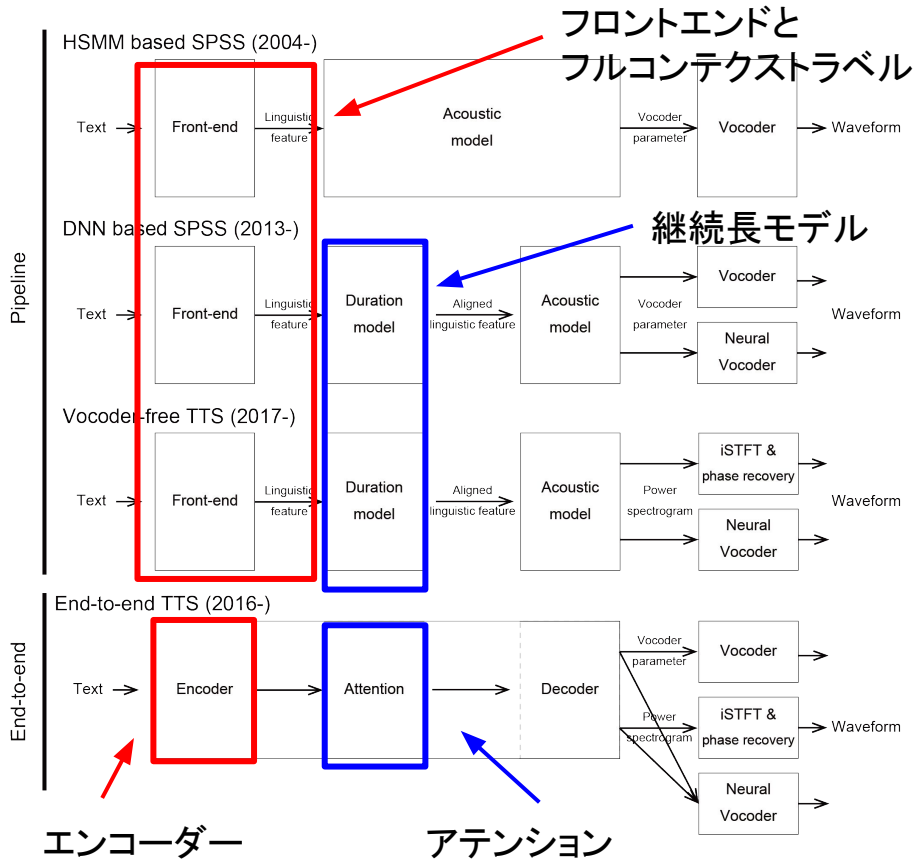
1) End-to-end TTSをピッチアクセント言語や声調言語に適用可能にする
2) TTSに特化したコンポーネントを導入することで、End-to-end TTSのアラインメントを安定化する



1. 導入：貢献

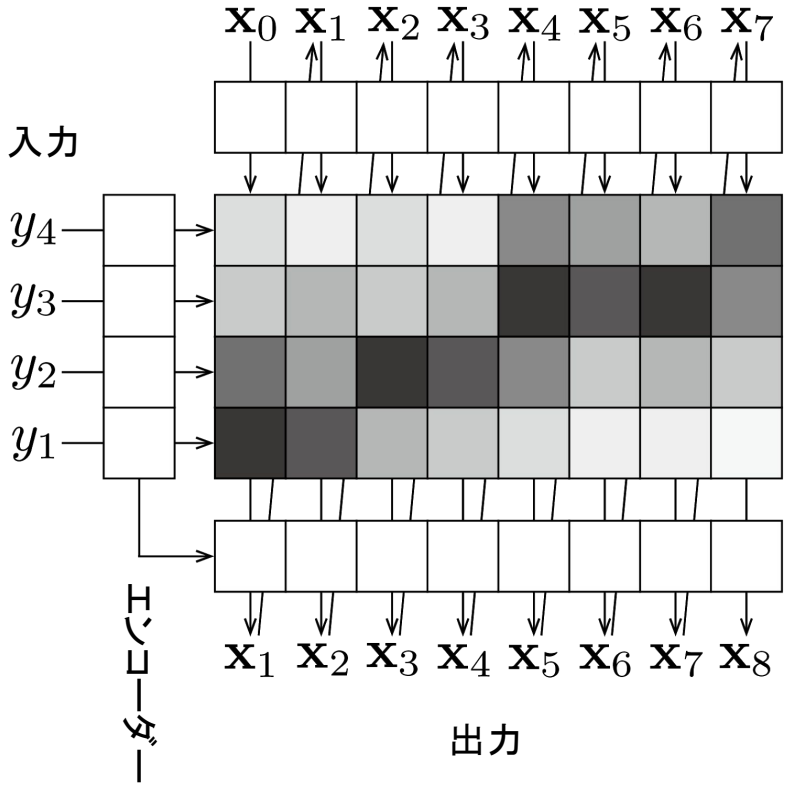
- 分析（3章）
  - End-to-endモデルのパラメーターサイズの大きさがアラインメントの安定性と発音解決に重要な要素。しかしピッチアクセントまでは解決できない。
  - Soft-attentionは致命的なアラインメントエラーを完全には解決できない。
- ピッチアクセント（4，5章）
  - ラベルを与えればEnd-to-end TTSは正しいピッチアクセントを反映した音声を予測できる。
  - ピッチアクセントを潜在変数として扱うEnd-to-end TTSの手法を提案。推論時にラベルを与えなくてもある程度正しいピッチアクセントを予測可能。
- 音韻継続長（6，7章）
  - 離散アラインメントを用いることで、単調増加なアラインメントを設計できるEnd-to-end TTSの手法を提案。致命的なアラインメントエラーを回避。
  - 音素継続長を潜在変数とするEnd-to-end TTS手法を提案。より効率的に単調増加なアラインメントを実現。

2. TTSフレームワーク



- パイプラインTTS
  - 複数のモデルを組み合わせてテキストから音声に変換する
  - フロントエンドがテキスト解析と言語モデルをもとにフルコンテクストラベルを生成
  - 継続長モデルが継続長をもとにラベルと音声のアラインメントを構築
- End-to-end TTS
  - 1つのモデルでテキストから音響特徴量へ変換を行う
  - エンコーダーがテキストから発音など音声に必要な情報を解決する
  - アテンションがテキストと音声との対応関係（アラインメント）をとる

2. 主要なEnd-to-end TTS手法の挙動



アテンション層

アテンション確率

$$p(z_t = u \mid \mathbf{x}_{t-1}, \mathbf{y}_u)$$

コンテクストベクトル

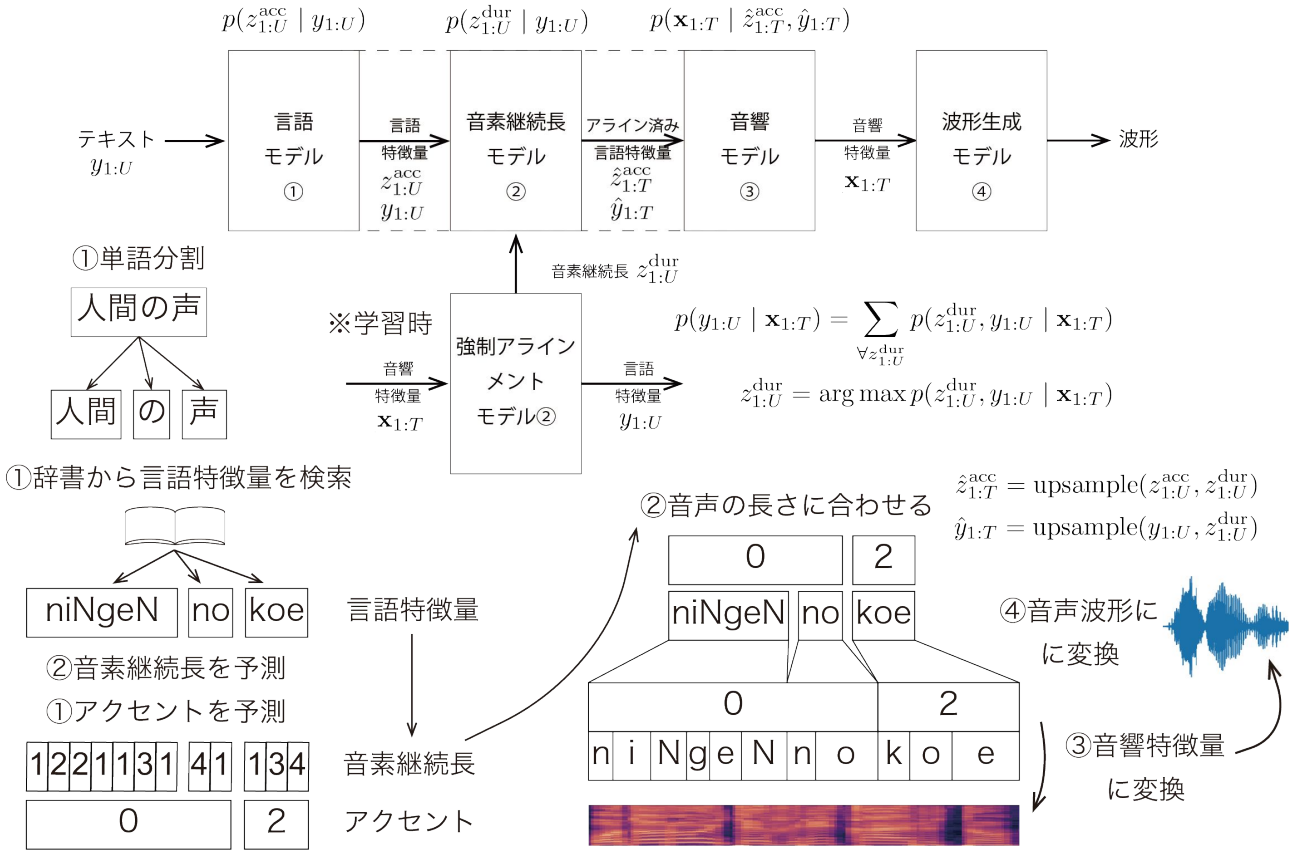
$$\mathbf{c}_t = \sum_{u=1}^{U} p(z_t = u \mid \mathbf{x}_{t-1}, \mathbf{y}_u)\mathbf{y}_u$$

$$p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{c}_t)$$

出力確率

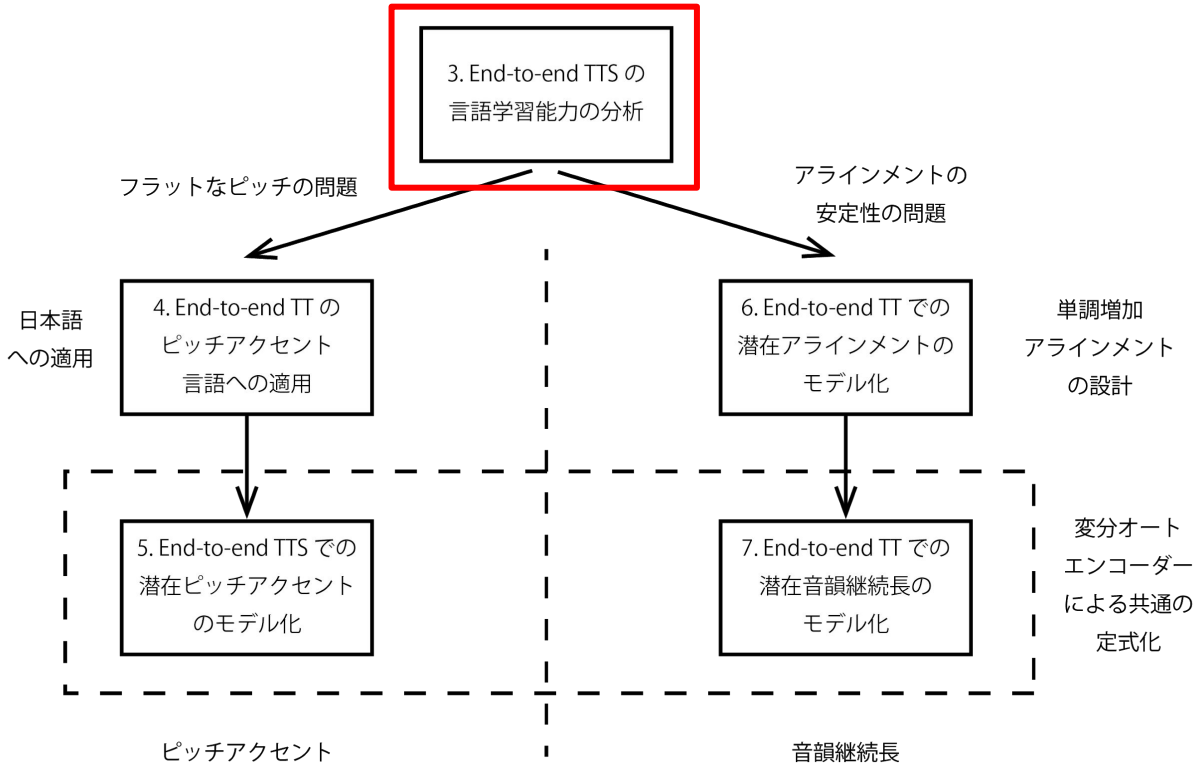
- 入力をエンコーダーでエンコードする
- 出力フィードバックとエンコードされた入力とのアテンション確率を計算する
- アテンション確率でエンコードされた入力の期待値をとり、コンテクストベクトルとする
- コンテクストベクトルから出力確率を計算する

2. パイプラインTTSフレームワークの挙動

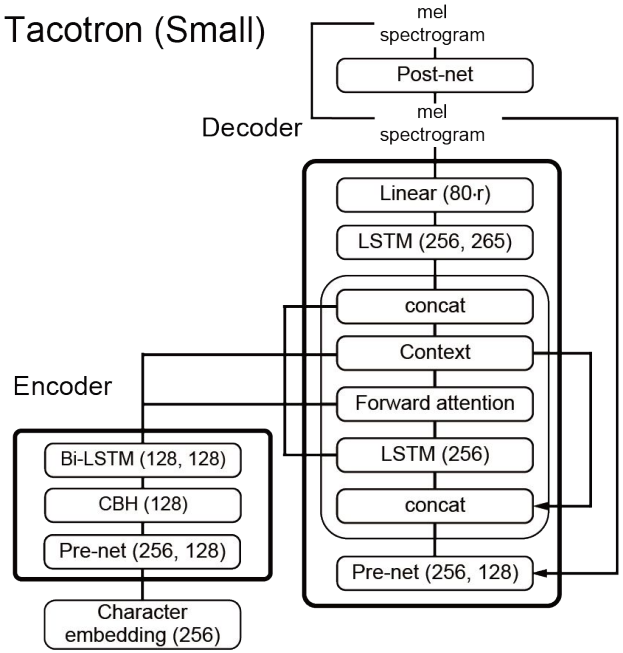


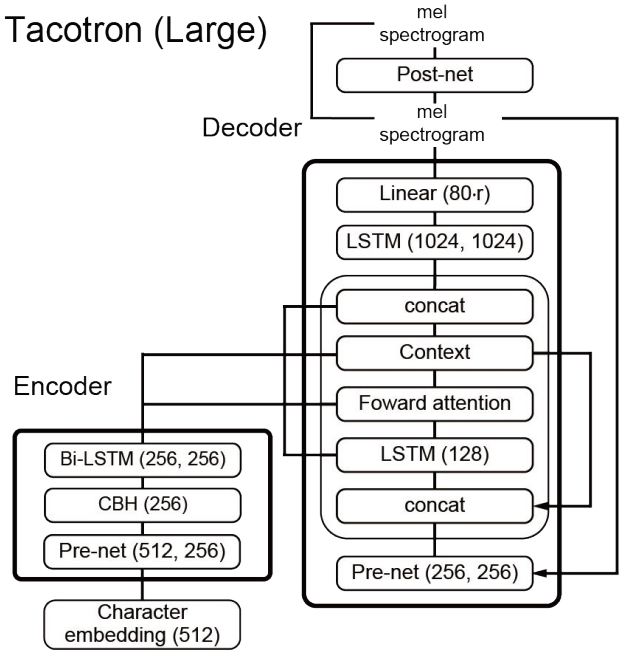
- 言語、音素継続長、音響、波形の独立したモデルからなる
- 各モデルの出力をパイプラインすることによって音声に変換する

3. End-to-end TTSの言語学習能力の分析



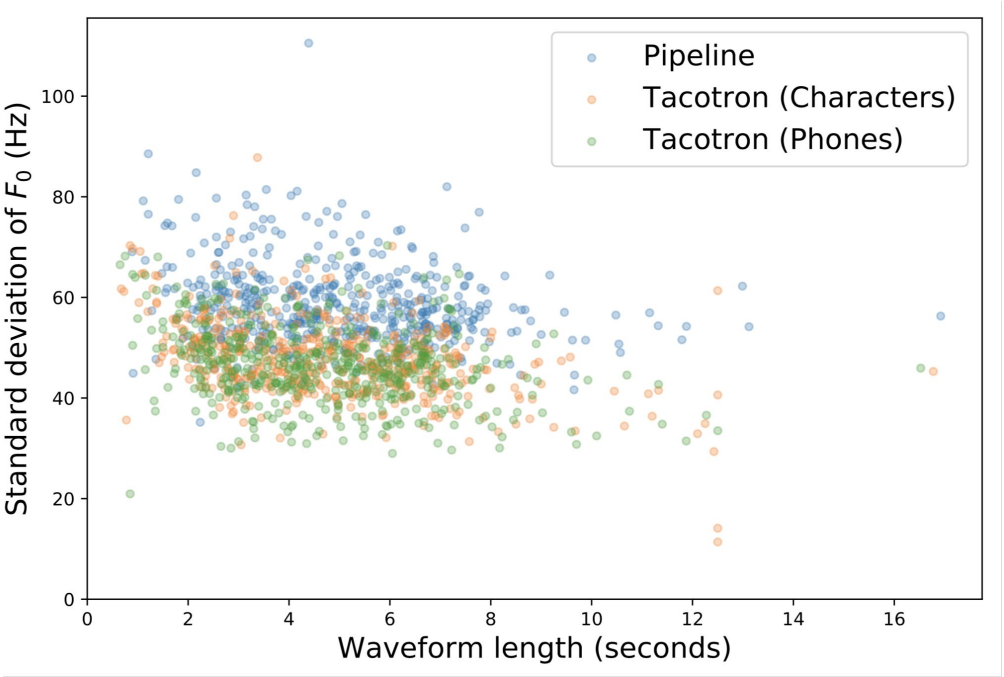
3. End-to-end TTSの代表的手法Tacotronを使った実験





- データ：Blizzard2011 (英語、単一女性話者、17h、12,092文)
- 手法：Tacotron
- パラメーターサイズ：
  - Small
  - Large
- 言語特徴量：
  - 文字列
  - 音素
- エンコーダー：
  - CNN
  - CBHL

3. End-to-end TTSの問題（1）: 平坦なピッチ



- End-to-end TTSは従来法のパイプラインに比べてピッチ変化が平坦な音声を予測する傾向にある。
- これが原因でEnd-to-end TTSはパイプラインに比べて低い自然性と評価された

3. End-to-end TTSの問題（2）：致命的アラインメントエラー

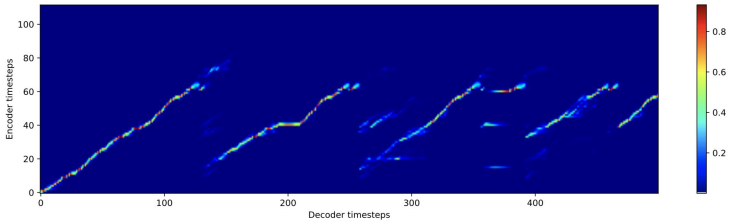
Para. size	Encoder	Self-attention	# Para. (1×10^6)	Alignment error rate (%) Ave.	1	2	3
Small	CBHL	-	7.9	7.6	6.2	8.6	9.0
		✓	12.1	17.9	10.0	10.4	33.2
	CNN	-	9.2	9.3	5.0	5.2	17.8
		✓	9.9	15.6	13.0	16.6	17.2
Large	CBHL	-	36.7	1.0	**0.6**	0.8	1.6
		✓	47.6	0.6	**0.2**	0.6	1.0
	CNN	-	27.2	1.1	**0.2**	1.4	1.6
		✓	38.1	1.0	0.8	1.0	**1.2**

(a) Systems using phone as input

Para. size	Encoder	Self-attention	# Para. (1×10^6)	Alignment error rate (%) Ave.	1	2	3
Small	CBHL	-	11.3	14.9	7.4	8.6	28.8
		✓	12.1	23.4	18.8	21.8	29.6
	CNN	-	9.2	11.1	6.6	10.8	15.8
		✓	9.9	18.0	15.0	16.8	22.2
Large	CBHL	-	36.7	1.1	0.8	**1.0**	1.6
		✓	47.6	0.8	0.4	**1.0**	1.0
	CNN	-	27.2	0.5	**0.2**	0.6	0.6
		✓	38.1	0.7	**0.4**	0.6	1.0

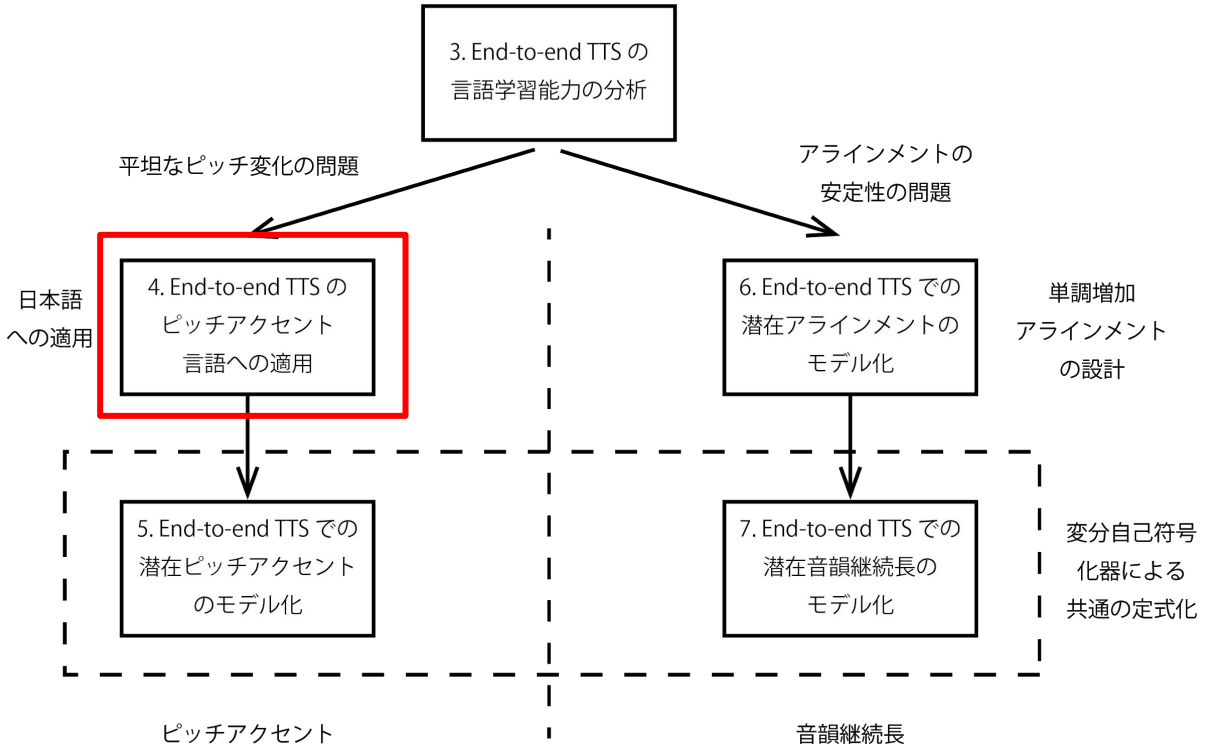
(b) Systems using character as input

- モデルパラメータサイズを十分大きくしても、致命的なアラインメントエラーはなくならない。
- アラインメント法に使用しているソフトアテンションが柔軟すぎることが原因



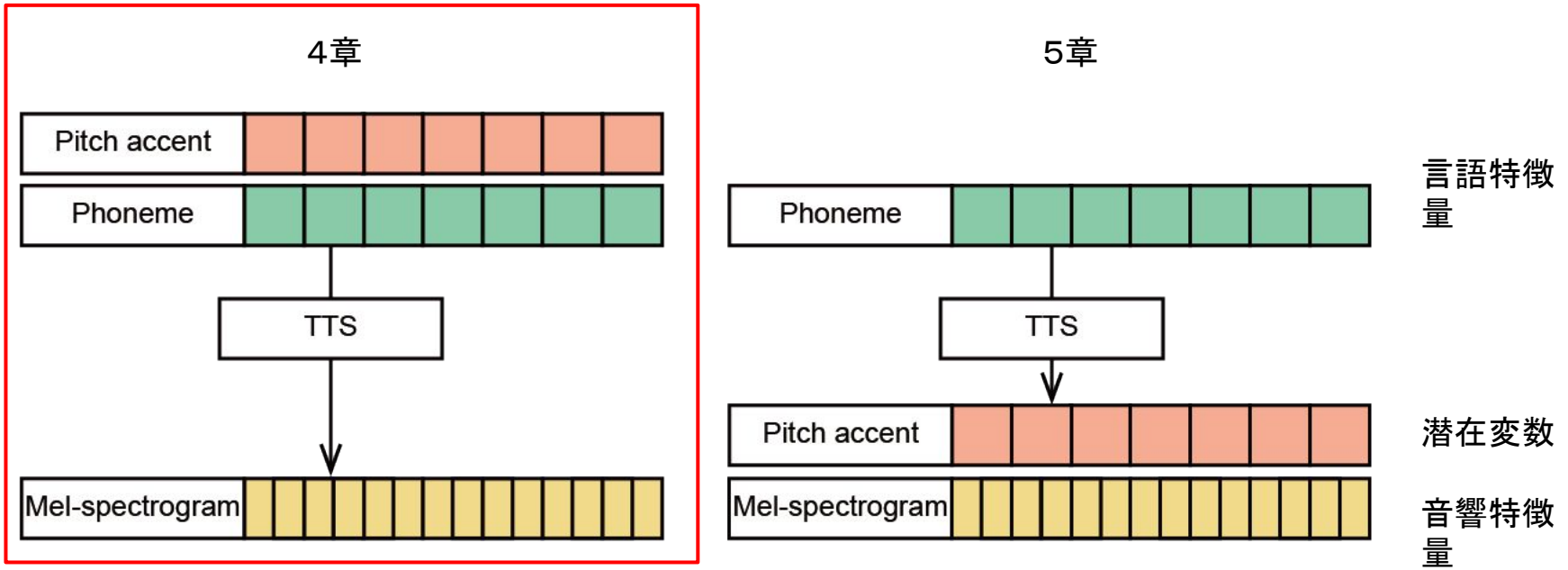
アラインメントエラーの例

4. End-to-end TTSのピッチアクセント言語への適用

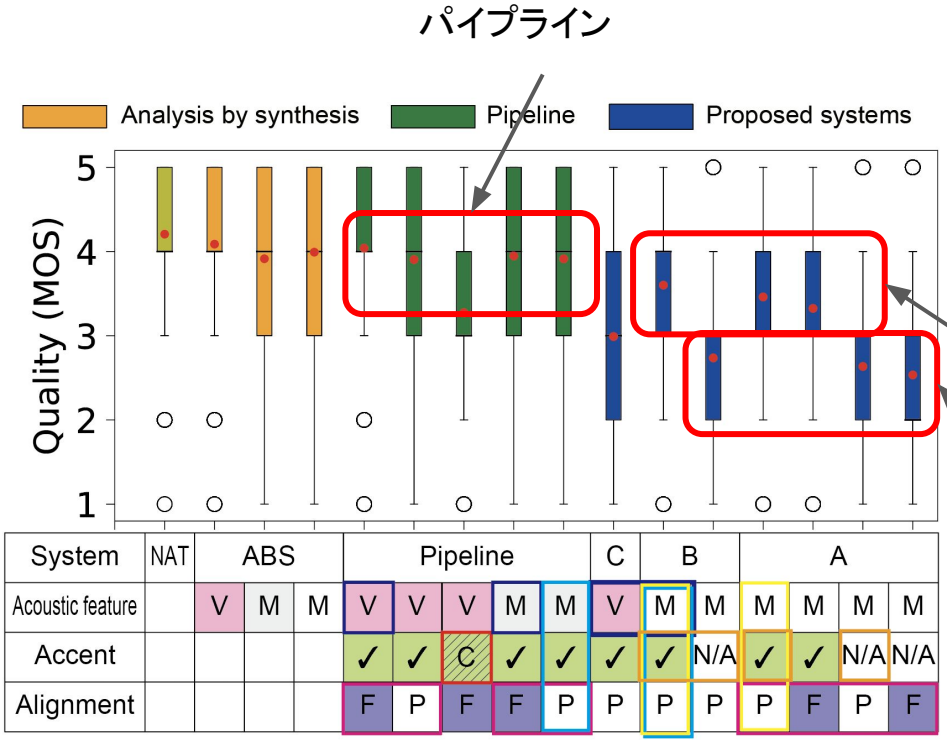


4. 日本語End-to-end TTSでのピッチアクセントの扱い

- 漢字かな交じりテキストの代わりに音素を用いる。
- ピッチアクセントラベル（アクセント型）を追加の入力として与える。



4. 実験：アクセントラベルの有無



System	NAT	ABS			Pipeline					C	B		A			
Acoustic feature		V	M	M	V	V	V	M	M	V	M	M	M	M	M	M
Accent					✓	✓	C	✓	✓	✓	✓	N/A	✓	✓	N/A	N/A
Alignment					F	P	F	F	P	P	P	P	P	F	P	F

音響特徴量
- M M メルスペクトログラム（12.5 / 5ms シフト長）
- V ボコーダーパラメーター

アクセント型ラベル
- ✓ あり　N/A なし
- C ノイズ（アクセント核位置をランダムにずらす）

アラインメント
- P 予測
- F 強制

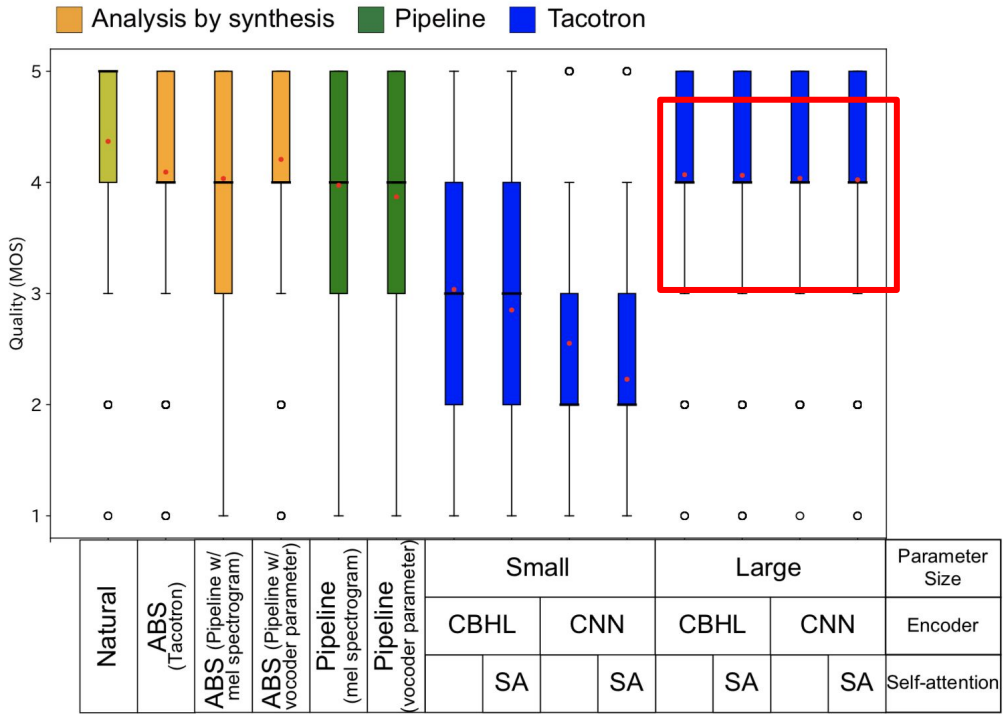
アクセントラベルあり

アクセントラベルなし

【結果】
- アクセントラベルがないと正しいアクセントの発話にはならない
- 提案手法は従来法のパイプラインに及ばない

音声サンプル https://nii-yamagishilab.github.io/samples-self-attention-tacotron/

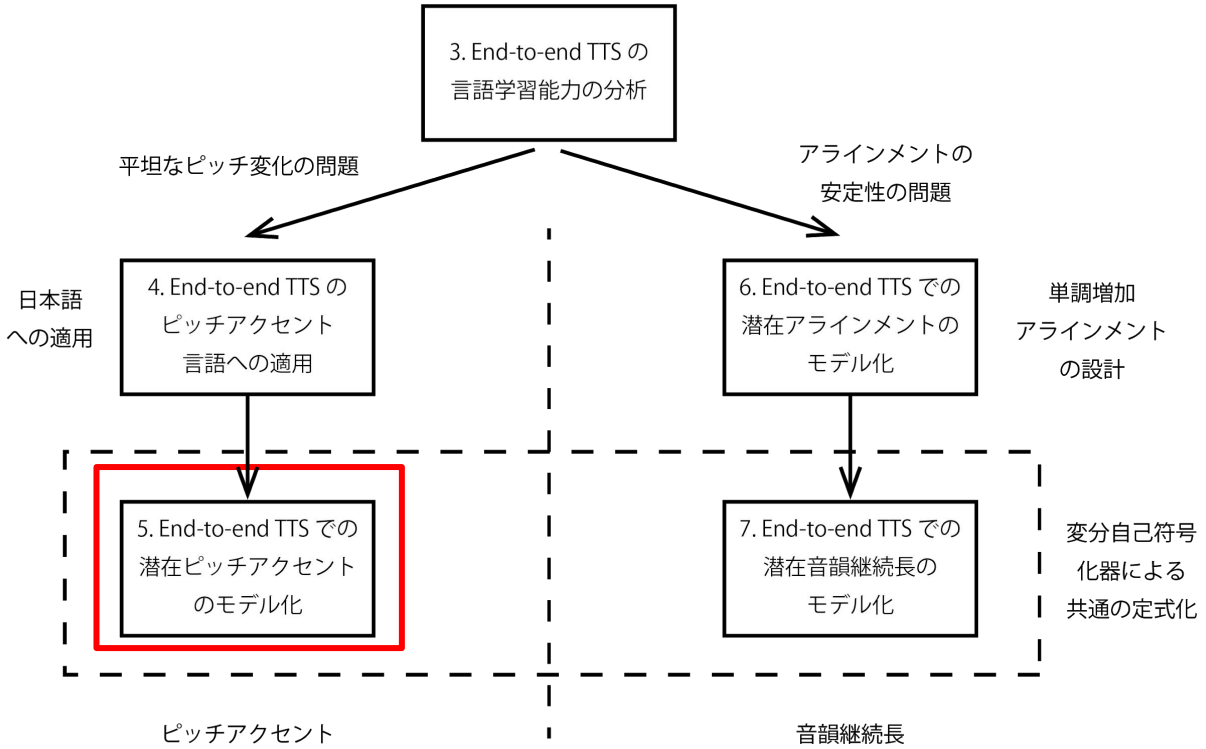
4. 実験：モデルパラメーターサイズの影響



- モデルパラメータサイズを十分大きくすれば、複雑な言語特徴量を使わずに、音素とアクセント型のみで非常に高い自然性を達成できる。

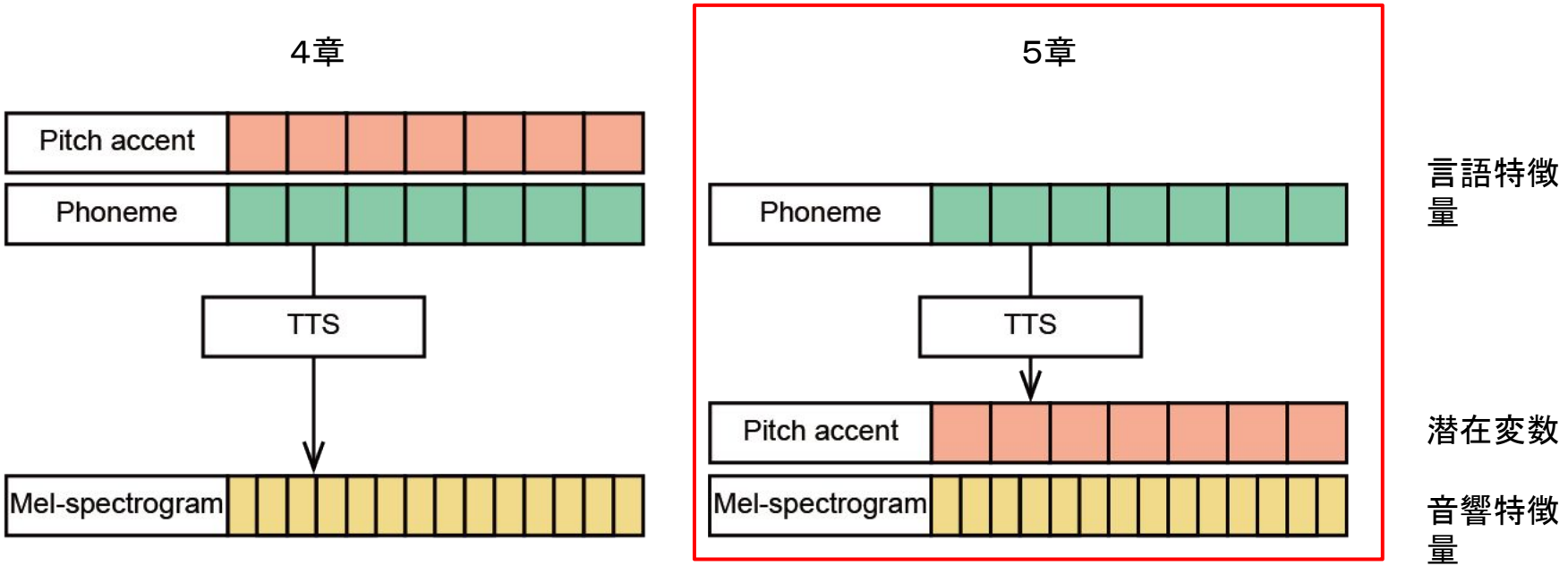
音声サンプル https://nii-yamagishilab.github.io/yasuda-csl-tacotron-audio-samples/

5. End-to-end TTSでの潜在ピッチアクセントのモデル化

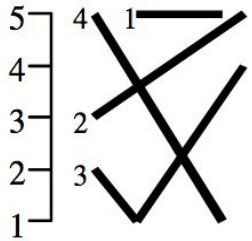


5. 潜在ピッチアクセントのモデル化

“ピッチアクセントラベルを入力として与える”（4章）から、”ピッチアクセントを音声とともに予測する”（5章）に拡張する。

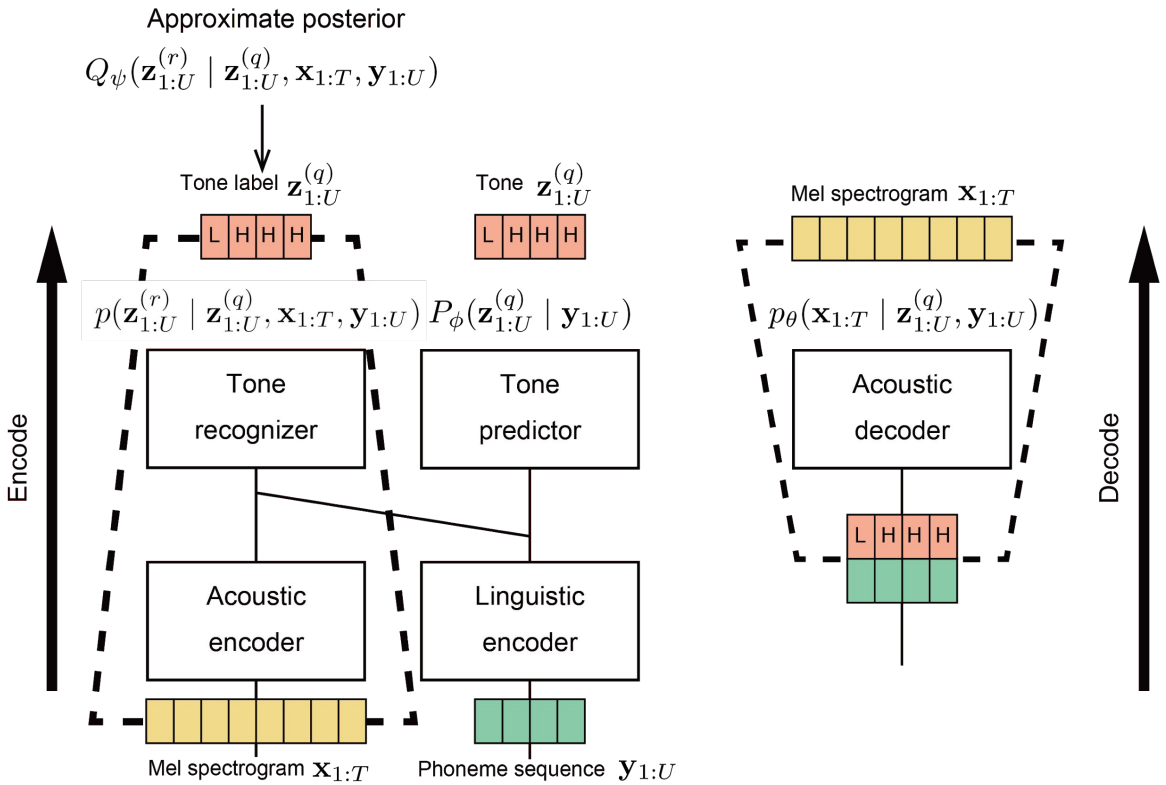


5. ピッチアクセントのトーン表現

Tone Contour	Tonal Feature
	H-H (H) L-H (R) L-L (L) H-L (F)

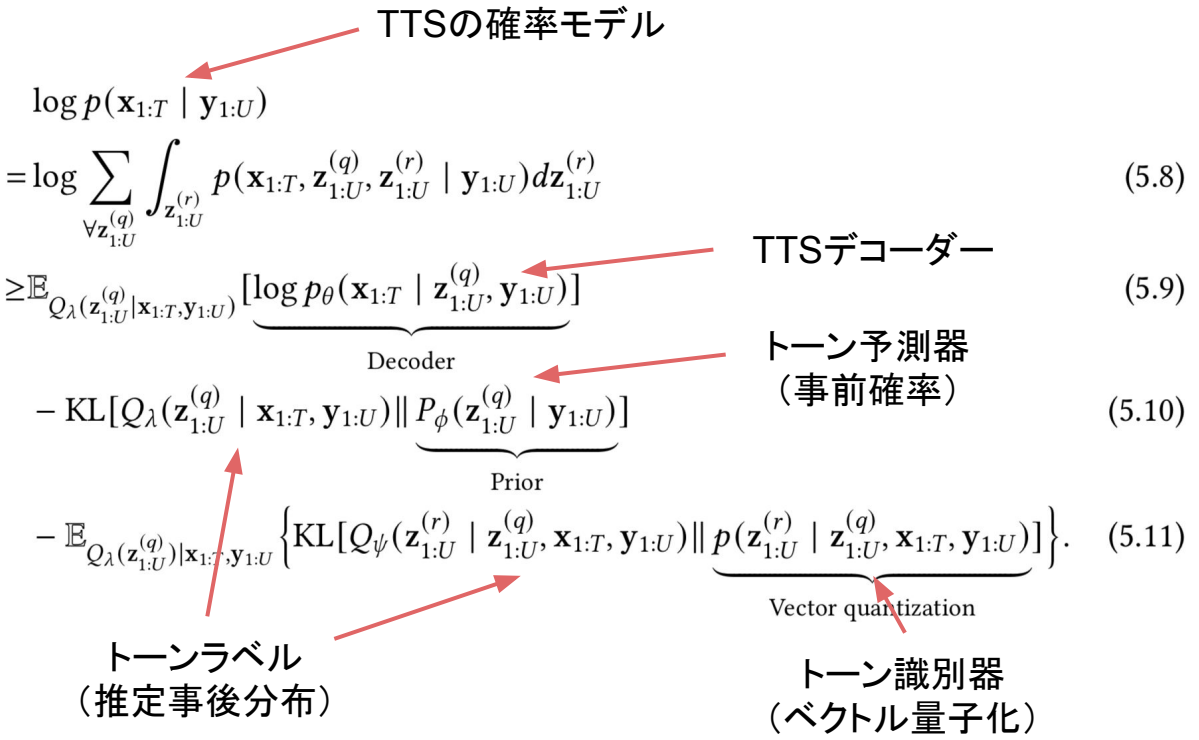
- ピッチ変化の抽象表現であるトーンで音節単位のピッチアクセントを表現する
- 日本語はX-JToBI、中国語はC-ToBIというToBIの拡張形式がある
- 以下のパターンがある:
  - LL (low)
  - HH (high)
  - LH (rising)
  - HL (falling)
  - L (neutral low)
  - H (neutral high)
- 日本語ではHL, L, Hの3種類
- 中国語ではHH, LH, LL, HL, H, Lの6種類

5. 変分自己符号化器(VAE)による潜在ピッチアクセントのモデル化



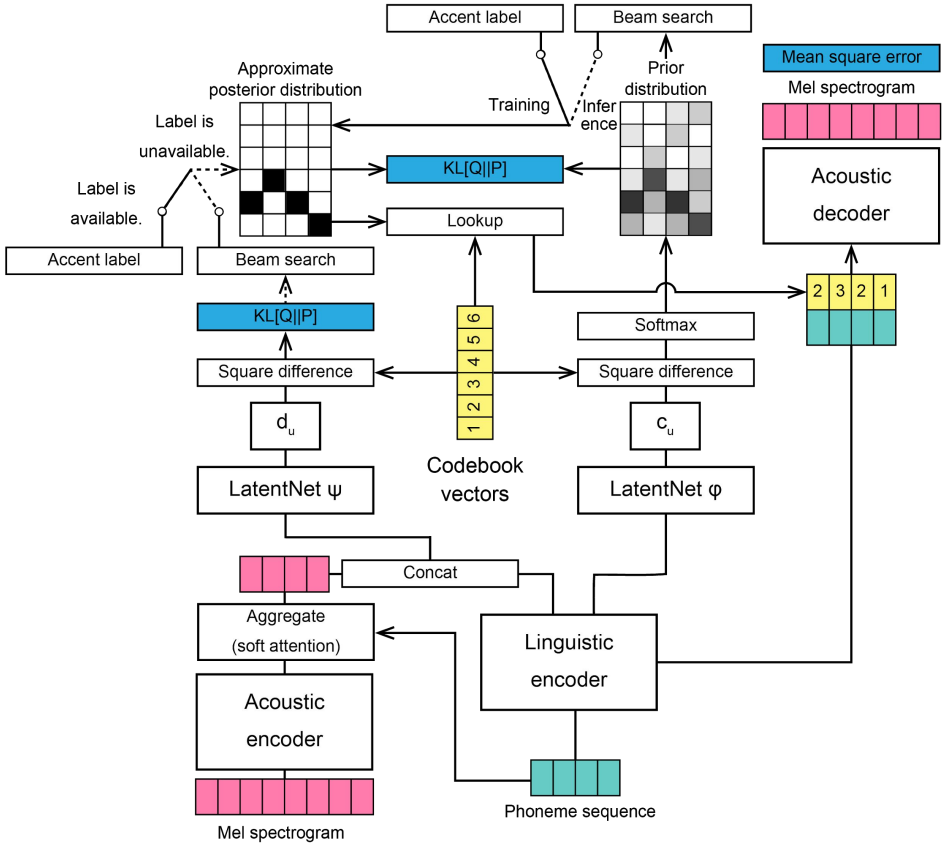
- トーンを離散潜在変数($\mathbf{z}^{(q)}_{1:U}$, $\mathbf{z}^{(r)}_{1:U}$)として導入
- トーンラベルで推定事後確率を定義
- トーン認識器をVAEのエンコーダーとして用いる
- トーン予測器をVAEの事前確率モデルとして用いる
- TTSのデコーダーをVAEのデコーダーとして用いる

5. 条件付きVQ-VAEの変分下限



- TTSは条件付き確率モデル（x: 音声, y: 言語特徴量）
- 周辺確率を変分下限で近似
- VQ-VAEの理論的解釈に基づき展開すると、以下のモジュールをTTSに組み込める
  - TTSデコーダー
  - トーン予測器
  - トーン識別器
  - トーンラベル

5. 提案システムのアーキテクチャ



- 共通
  a. 言語特徴量とアクセントをアテンションで音響特徴量に合わせる
  b. デコーダーで音響特徴量にデコード
- 学習時
  - アクセントラベルを用いて推定事後確率を定義
  - アテンションで音響特徴量を言語特徴量に合わせる
  - エンコーダーで音響特徴量と言語特徴量をアクセント潜在表現にエンコード
- 予測時
  - アクセント予測器からアクセント潜在変数をサンプル

5. 実験

- 日本語データ
  - ATR Ximera (50 h)
  - トーンラベル: X-JToBI (HL, H, L)
  - 句境界ラベル: NA, アクセント句, 呼気段落
- 中国語データ
  - Standard Mandarin (12 h)
  - トーンラベル: C-ToBI (LL, HH, LH, HL, H, L)
- システム
  - 学習時と推論時の条件を変えた5システム
  - U-, S-, SS, PP, M-
- 評価
  - トーンエラー率（TER）
  - リスニングテスト
    - 自然性（5段階MOS）
    - アクセントの正しさ（4段階 MOS）

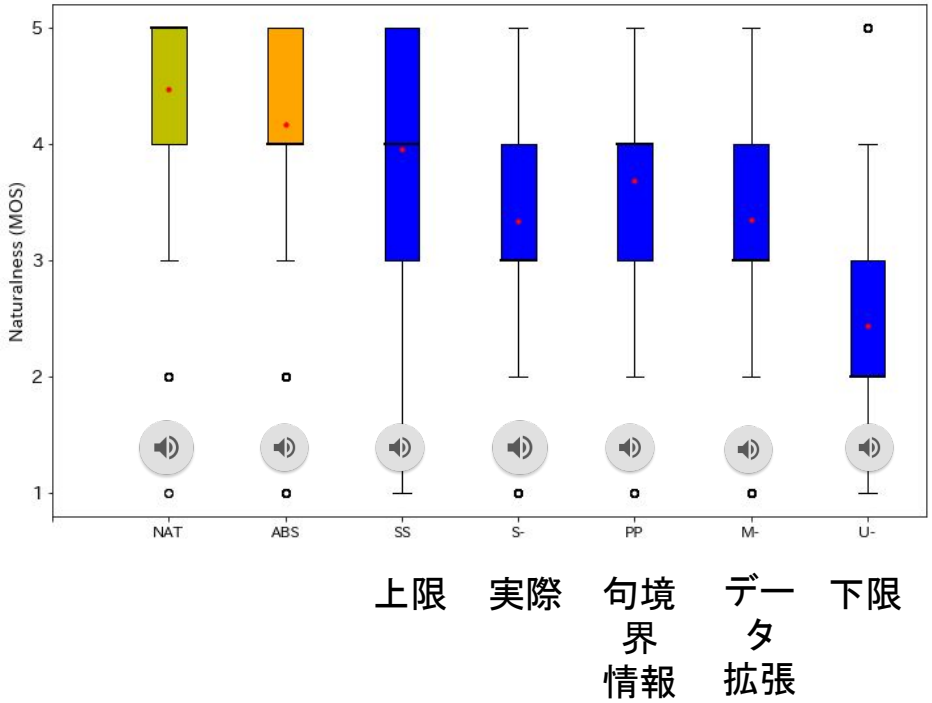
	System	Accent label	
		Training	Inference
下限	U−	-	-
	US	-	Speech
実際	S−	Tone	-
上限	SS	Tone	Speech
句境界情報	PP	Tone & Phrase	Phrase
データ拡張	M−	Tone	-

5. トーンエラー率（TER）

	System	Accent label Training	Accent label Inference	TER (%) Test
下限	U−	-	-	-
	US	-	Speech	56.7
実際	S−	Tone	-	25.1
上限	SS	Tone	Speech	6.1
句境界情報	PP	Tone & Phrase	Phrase	13.5
データ拡張	M−	Tone	-	23.0

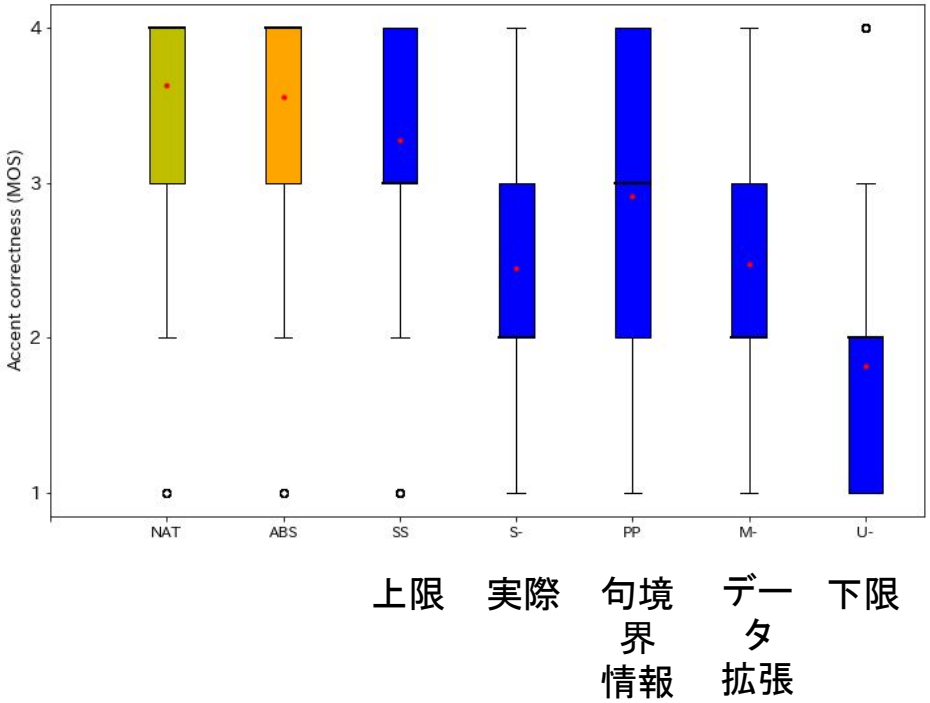
- ラベルを与えない場合潜在変数はピッチアクセントを表さない
- ラベルを与えれば、潜在空間にピッチアクセントを捉えられる
- トーン識別性能は高く、VAEとしての再構築性能は高い
- 句境界情報は役に立つ
- 中国語でのデータ拡張は効果なし

5. 結果1）自然性



- 推論時に音声を与えた場合、高い自然性
- 学習時のみラベルを使った場合、そこそこ高い自然性
- 句境界情報は自然性を向上する
- 中国語でのデータ拡張は効果なし
- ラベルを使わない場合は自然性が低い

5. 結果2）アクセントの正しさ



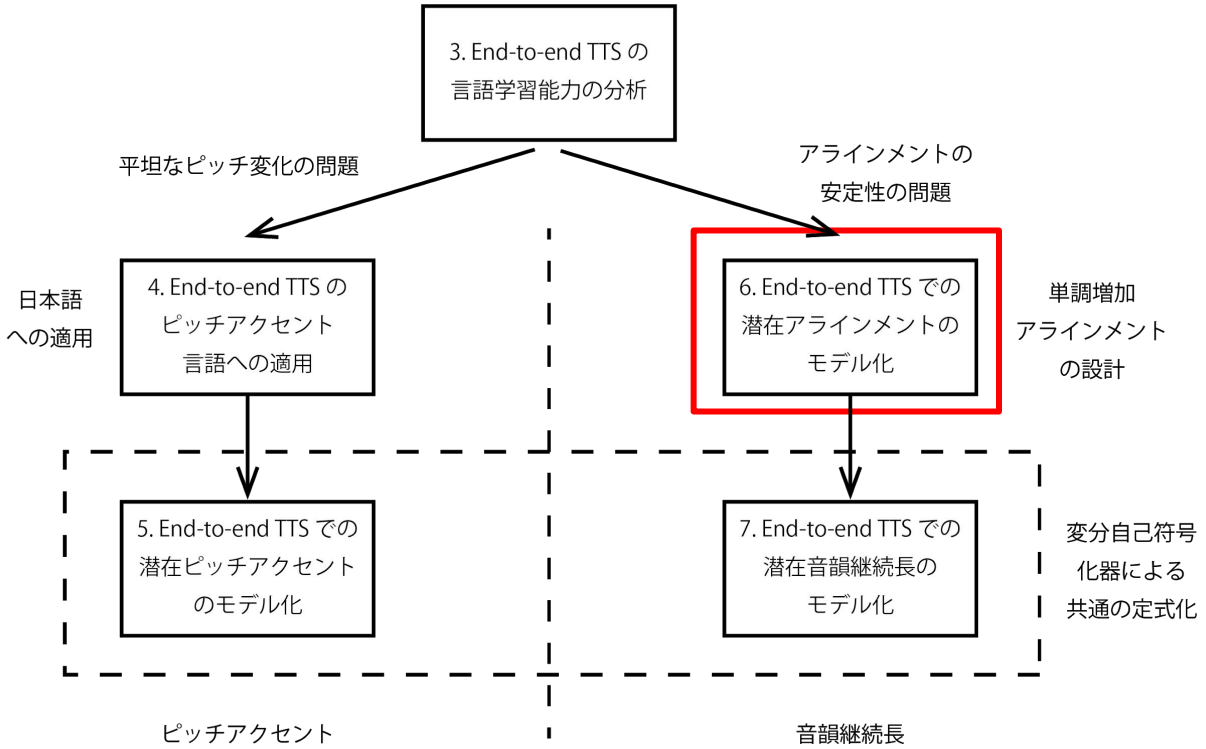
- 推論時に音声を与えた場合、正しいアクセント
- 学習時のみラベルを使った場合、アクセントの正しさは中間
- 句境界情報はアクセントの正確性を向上する
- 中国語でのデータ拡張は効果なし
- 学習時にラベルを使わない場合はアクセントは誤っている

8. 結論：解決した問題

ピッチアクセント

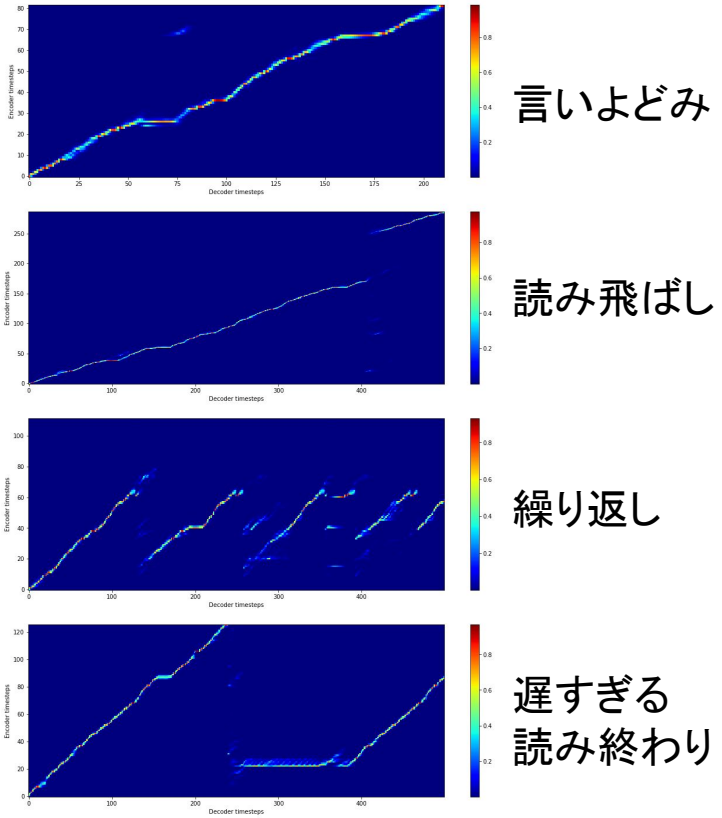
- End-to-end法ではピッチアクセント言語や声調言語におけるピッチアクセントの扱いが確立していない。
  - アクセント型を追加の入力として与えれば、End-to-end TTSは正しいアクセントを反映した音声を生成できる。
  - 十分大きなモデルパラメータサイズを用いれば、ピッチアクセント言語に対し音素とアクセント型のみの言語特徴量でEnd-to-end TTSは非常に高い自然性を達成できる。
- ピッチアクセントの予測は別のモデルによってなされることを前提としており、End-to-end法自体には取り込まれていない。
  - 条件付きVQ-VAEを用いることで、ピッチアクセントモデルを End-to-end TTSに組み込むことができる手法を提案。
  - 推論時にピッチアクセントラベルに依存せずに、そこそこ良いトーンエラー率と自然性を達成できる。

6. End-to-end TTSでの潜在アラインメントのモデル化

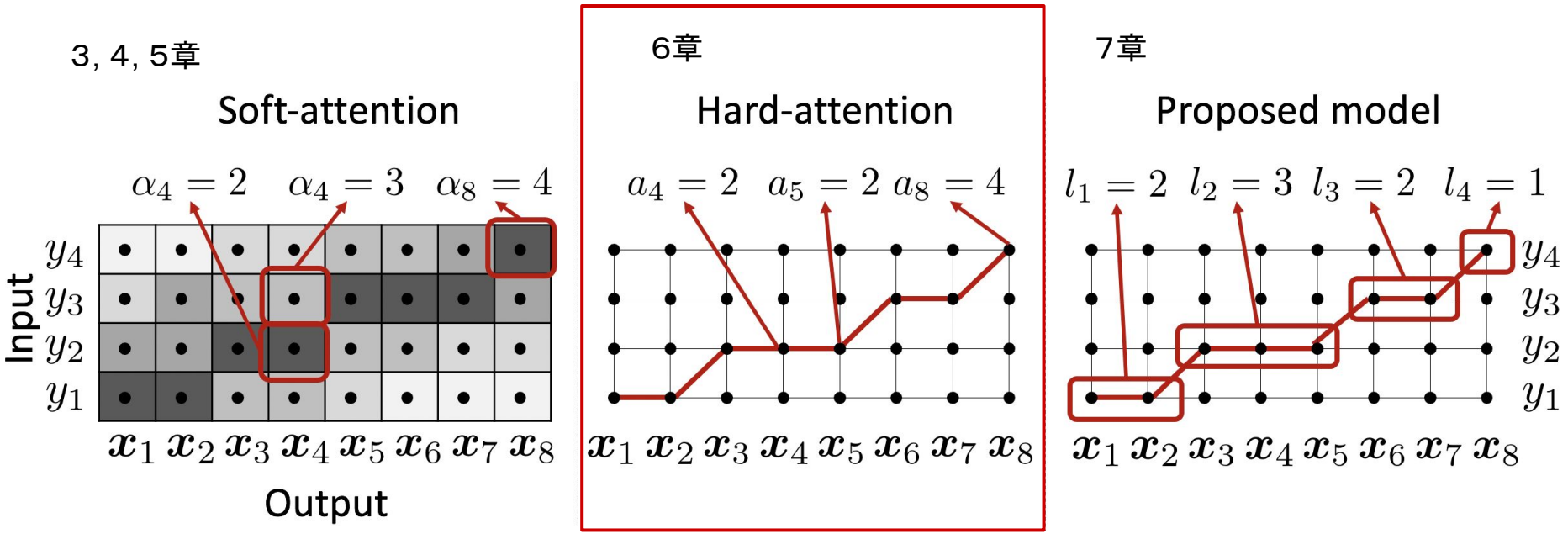


6. ソフトアテンションの問題点：致命的なアラインメントエラー

- 音声合成において，アラインメントは**単調増加**でなければならない
- ソフトアテンションは**柔軟すぎる**ので，**明らかに誤ったアラインメント**も予測することがある
- 推論の**停止基準は明確でなく**，予測する必要がある

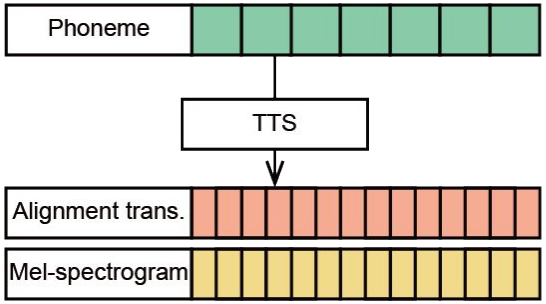


6. 潜在変数の設計



- 状態遷移を潜在変数として用い、アラインメントを単調増加にする
- 状態遷移を2値の離散変数とする (Emit: 留まる, Shift: 進む)

6. 単調増加離散アラインメントを用いるSSNT-TTS



Linguistic features $\mathbf{y}_{1:U}$

$$p(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:U}) = \sum_{\forall \mathbf{z}_{1:T}} p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T} \mid \mathbf{y}_{1:U})$$

$$\approx \sum_{\forall \mathbf{z}_{1:T}} \prod_{t=1}^{T} p(z_t \mid z_{t-1}, \mathbf{x}_{t-1}, \mathbf{y}_{1:U}) p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, z_t, \mathbf{y}_{1:U})$$

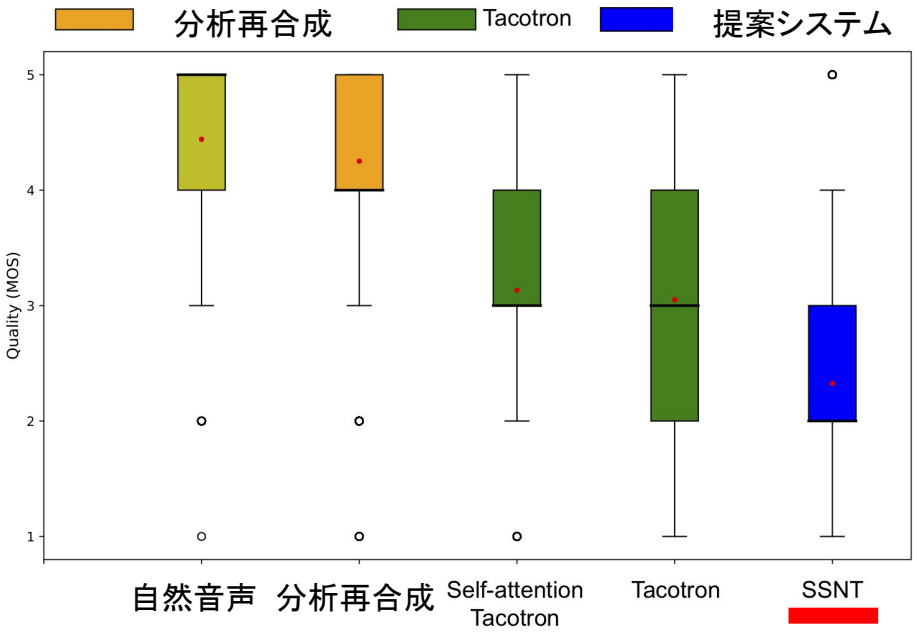
Acoustic features $\mathbf{x}_{1:T}$

Marginalization of latent alignment variable

Alignment probability

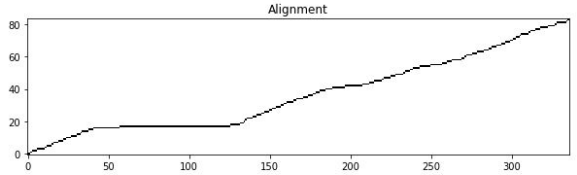
Output probability

6. SSNT-TTSの結果と課題

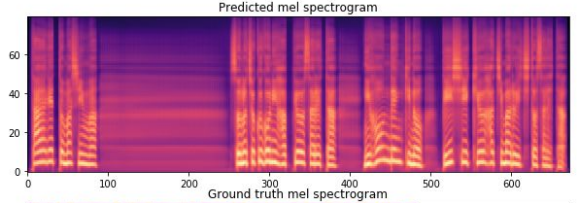


- 別の種類のアラインメントエラーがみられた
  - 音素継続長の過剰予測
  - 音素継続長の過少予測

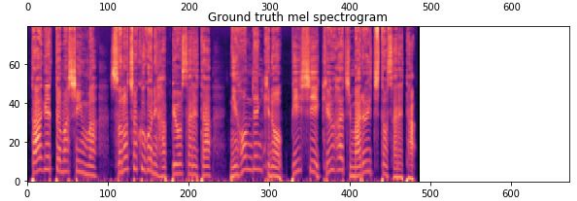
予測アラインメント



予測スペクトル



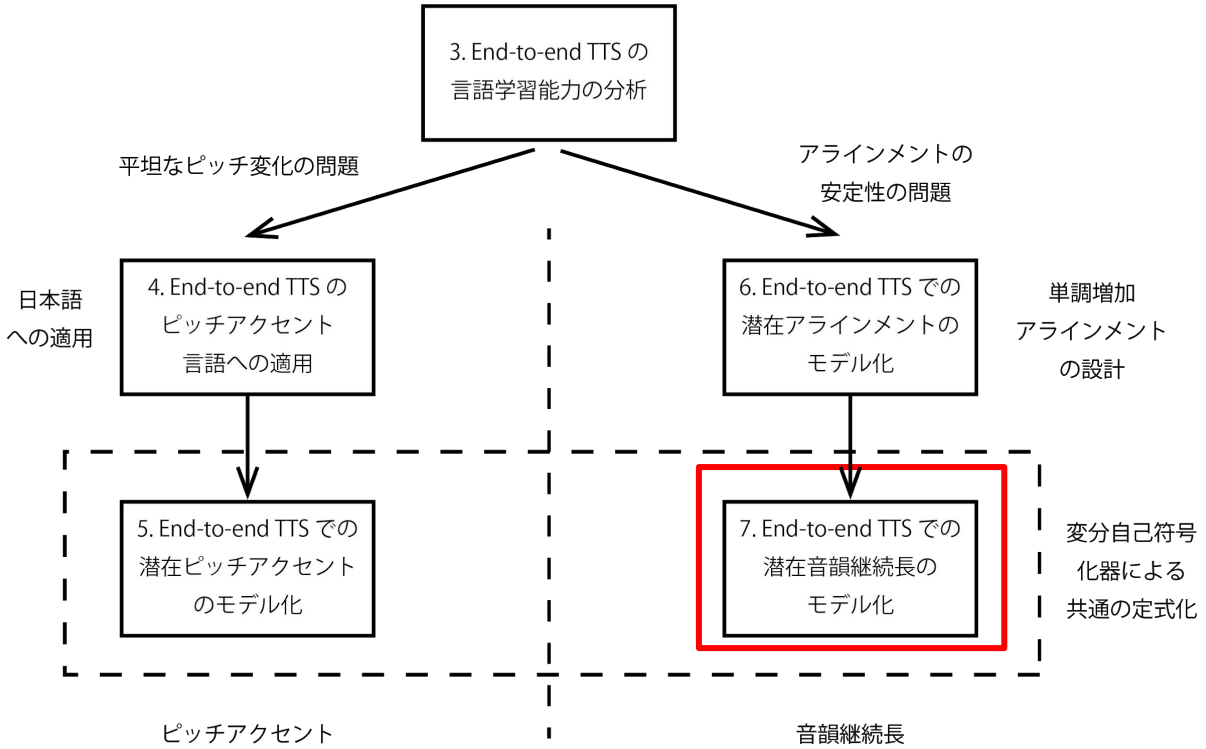
正解スペクトル



※アクセントラベルなし

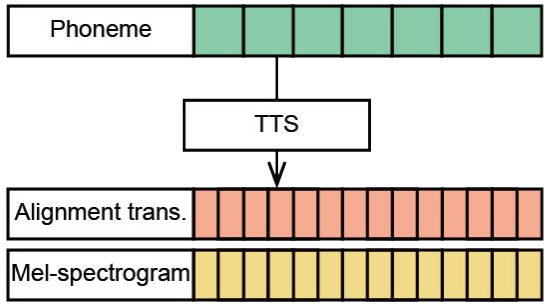
音声サンプル https://nii-yamagishilab.github.io/samples-ssnt/

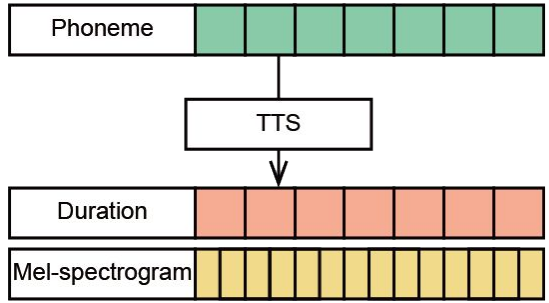
7. End-to-end TTSでの潜在音韻継続長のモデル化



7. 音声合成における音素継続長

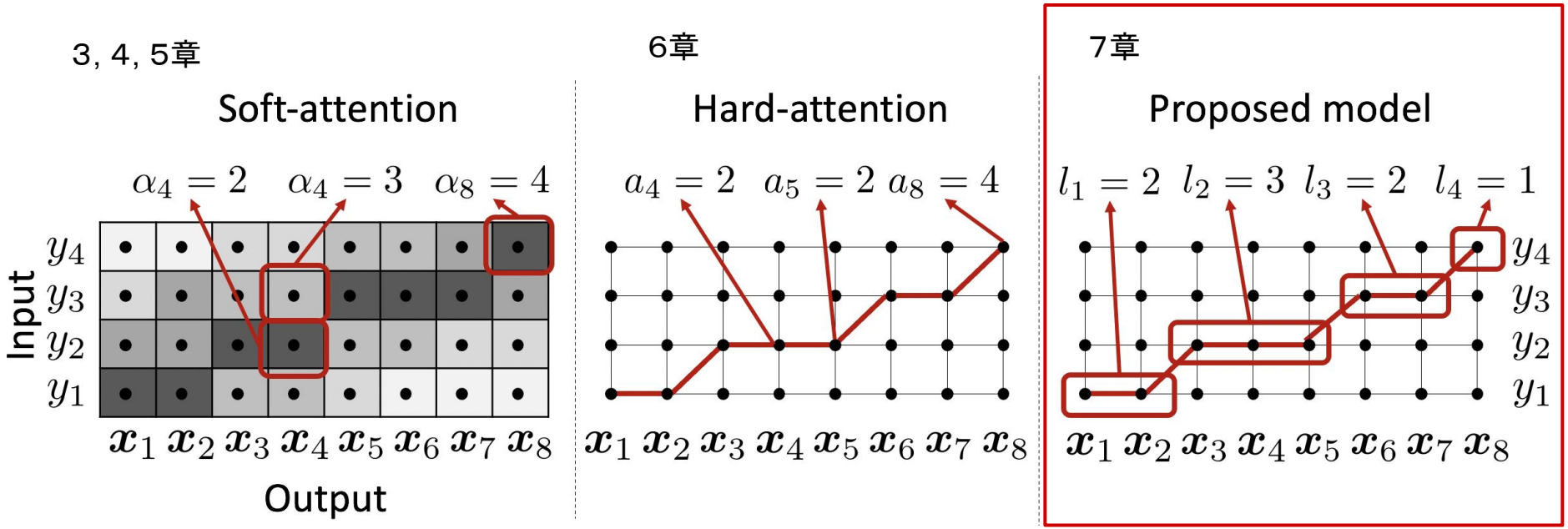
- 音声生成の観点
  - 各音素は“内在継続長”をもつと言われている
- エンジニアリングの観点
  - アラインメントは音素継続長で表現したほうが効率的
    - 音素継続長は単調増加アラインメントを保証
    - 系列長が短い（状態遷移は出力レベルなのに対し、音素継続長は入力レベル）





- 提案システムのデザイン
  - 音素継続長を潜在変数とする
  - パイプラインTTSで用いられる強制アラインメントと継続長モデルをEnd-to-end TTSに導入する
  - 上記を実現するために変分自己符号化器（ VAE）を使う

7. 潜在変数の設計

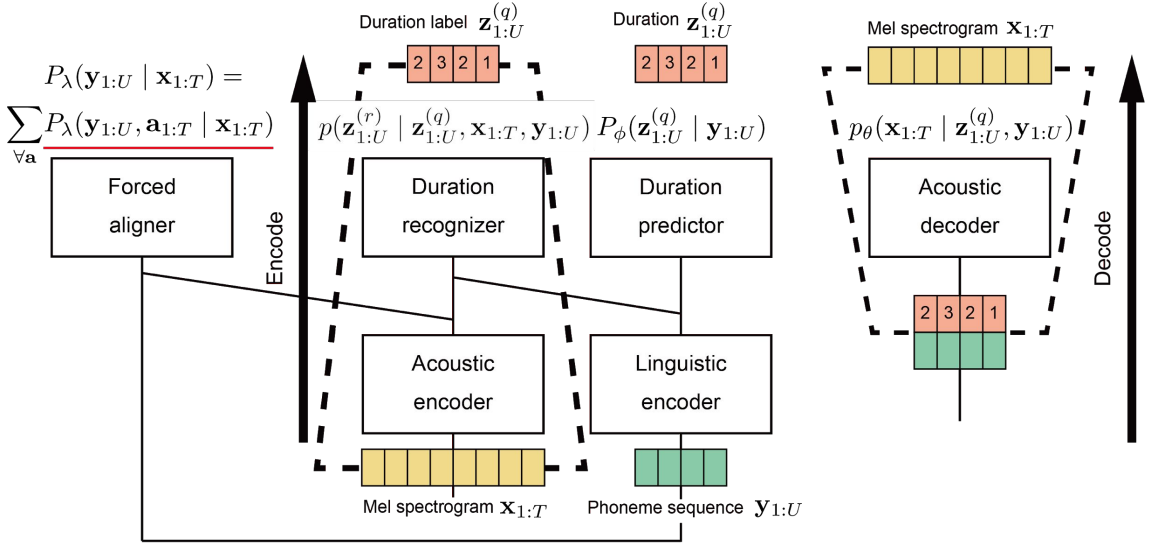


- 音素継続長を潜在変数として用いる
- 音素継続長を離散変数とする (音響特徴量のフレーム数)

7. 変分自己符号化器(VAE)による潜在継続長のモデル化

$$\mathbf{z}^{(q)}_{1:U} = \arg\max_{\mathbf{a}_{1:T}} P_\lambda(\mathbf{y}_{1:U}, \mathbf{a}_{1:T} \mid \mathbf{x}_{1:T})$$

Approximate posterior $\sim Q_\psi(\mathbf{z}^{(r)}_{1:U} \mid \mathbf{z}^{(q)}_{1:U}, \mathbf{x}_{1:T}, \mathbf{y}_{1:U})$



- 継続長を離散潜在変数 (, $\mathbf{z}^{(q)}_{1:U}$) $\mathbf{z}^{(r)}_{1:U}$として導入
- 強制アラインメントから継続長をサンプル
- 継続長サンプルで推定事後分布を定義
- 継続長予測器をVAEの事前確率として用いる
- TTSのデコーダーをVAEのデコーダーとして用いる

7. 条件付きVQ-VAEの変分下限

$$\log p(\mathbf{x}_{1:T} \mid \mathbf{y}_{1:U})$$

TTSの確率モデル

$$= \log \sum_{\forall \mathbf{z}^{(q)}_{1:U}} \int_{\mathbf{z}^{(r)}_{1:U}} p(\mathbf{x}_{1:T}, \mathbf{z}^{(q)}_{1:U}, \mathbf{z}^{(r)}_{1:U} \mid \mathbf{y}_{1:U}) d\mathbf{z}^{(r)}_{1:U} \quad (7.1)$$

TTSデコーダー

$$\geq \mathbb{E}_{Q_\lambda(\mathbf{z}^{(q)}_{1:U})} \underbrace{[\log p_\theta(\mathbf{x}_{1:T} \mid \mathbf{z}^{(q)}_{1:U}, \mathbf{y}_{1:U})]}_{\text{Decoder}} \quad (7.2)$$

継続長予測器
（事前確率）

$$- \mathrm{KL}[Q_\lambda(\mathbf{z}^{(q)}_{1:U} \mid \mathbf{x}_{1:T}, \mathbf{y}_{1:U}) \| \underbrace{P_\phi(\mathbf{z}^{(q)}_{1:U} \mid \mathbf{y}_{1:U})}_{\text{Prior}}] \quad (7.3)$$

継続長認識器
（ベクトル量子化）

$$- \mathbb{E}_{Q_\lambda(\mathbf{z}^{(q)}_{1:U})} \left\{ \mathrm{KL}[Q_\psi(\mathbf{z}^{(r)}_{1:U} \mid \mathbf{z}^{(q)}_{1:U}, \mathbf{x}_{1:T}, \mathbf{y}_{1:U}) \| \underbrace{p(\mathbf{z}^{(r)}_{1:U} \mid \mathbf{z}^{(q)}_{1:U})}_{}] \right\}. \quad (7.4)$$

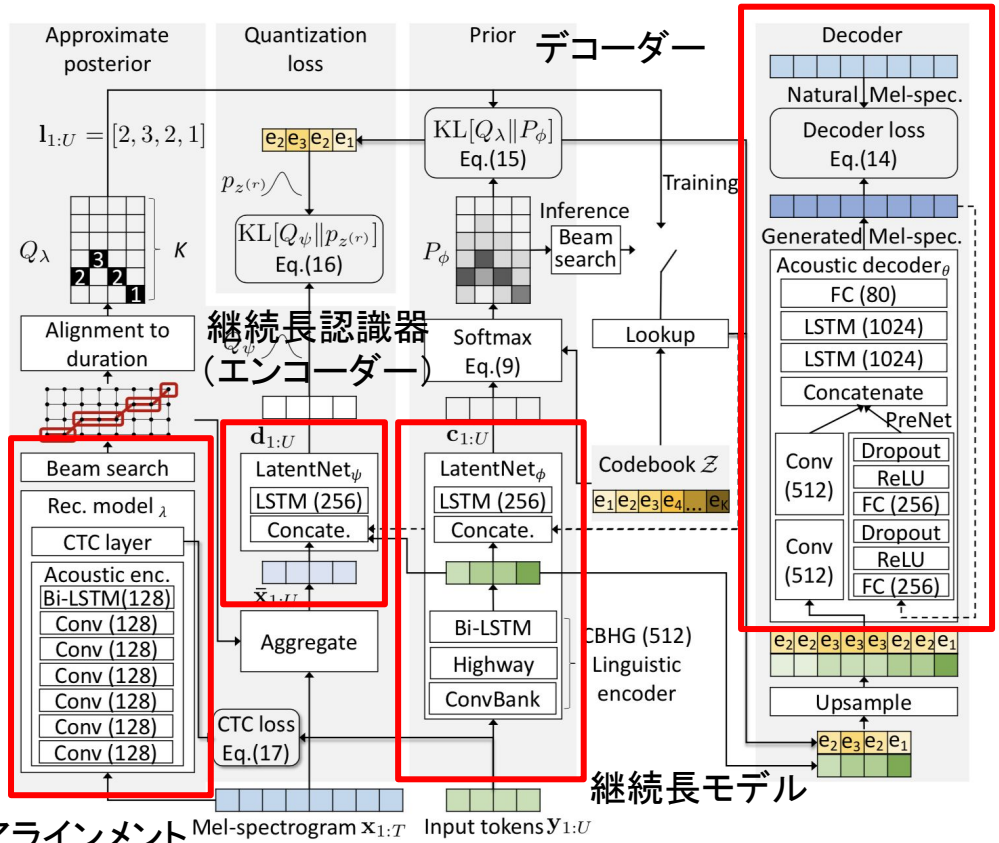
$$+ \gamma \log \sum_{\forall \mathbf{a}_{1:T}} P_\lambda(\mathbf{y}_{1:U}, \mathbf{a}_{1:T} \mid \mathbf{x}_{1:T}), \quad (7.17)$$

強制アラインメント

強制アラインメントの継続長サンプル
（推定事後分布）

- TTSは条件付き確率モデル（x: 音声, y: 言語特徴量）
- 周辺確率を変分下限で近似
- VQ-VAEの理論的解釈に基づき展開すると、以下のモジュールをTTSに組み込める
  - TTSデコーダー
  - 継続長予測器
  - 継続長認識器
  - 継続長サンプル
  - 強制アラインメント

7. 提案システムのアーキテクチャ



- 共通
  a. アラインメントを継続長に変換
  b. 継続長に基づき言語特徴量をアップサンプル
  c. デコーダーで音響特徴量にデコード
- 学習時
  - 強制アラインメントからアラインメントをサンプル
  - サンプルは継続長認識器と継続長モデルを学習するために使用
- 予測時
  - 継続長モデルからアラインメントをサンプル

7. 実験

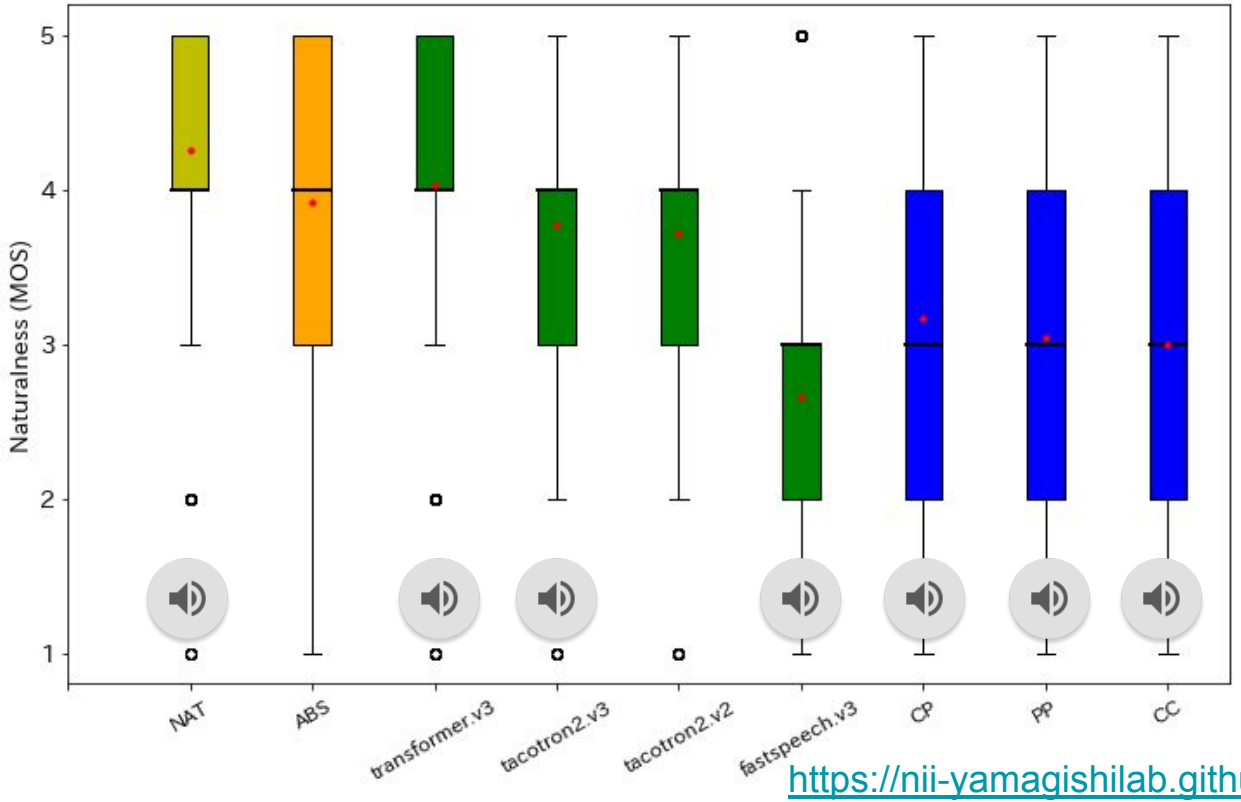
- データ: 英語 (LJSpeech, 24h)
- 提案システム
  - CP (TTSに文字列、強制アラインメントに音素)
  - PP (音素)
  - CC (文字列)
- ベースラインシステム:
  - Transformer TTS
  - Tacotron2 (v2, v3)
  - FastSpeech
  - ABS
  - Natural

End-to-end TTSの代表的な手法

継続長モデルを用いた代表的な手法

- 評価
  - 自然性に関するリスニングテスト（5段階 MOS）
  - 200被験者

7. 実験結果



- 自然性:
  - Natural
  - > End-to-end TTS
    - Transformer
    - Tacotron v3, v2
  - > 提案システム
    - CP, PP, CC
  - > 継続長モデルを用いたTTS
    - FastSpeech
- 提案システム
  - CP (character for TTS, phoneme for CTC) がもっともよい

https://nii-yamagishilab.github.io/sample-tts-latent-duration/

7. 提案システムの利点と問題点

- 利点
  - すべてのモジュールを同時に学習可能。学習回数は1回。
  - 自己回帰デコーダーによる強力な音響モデル

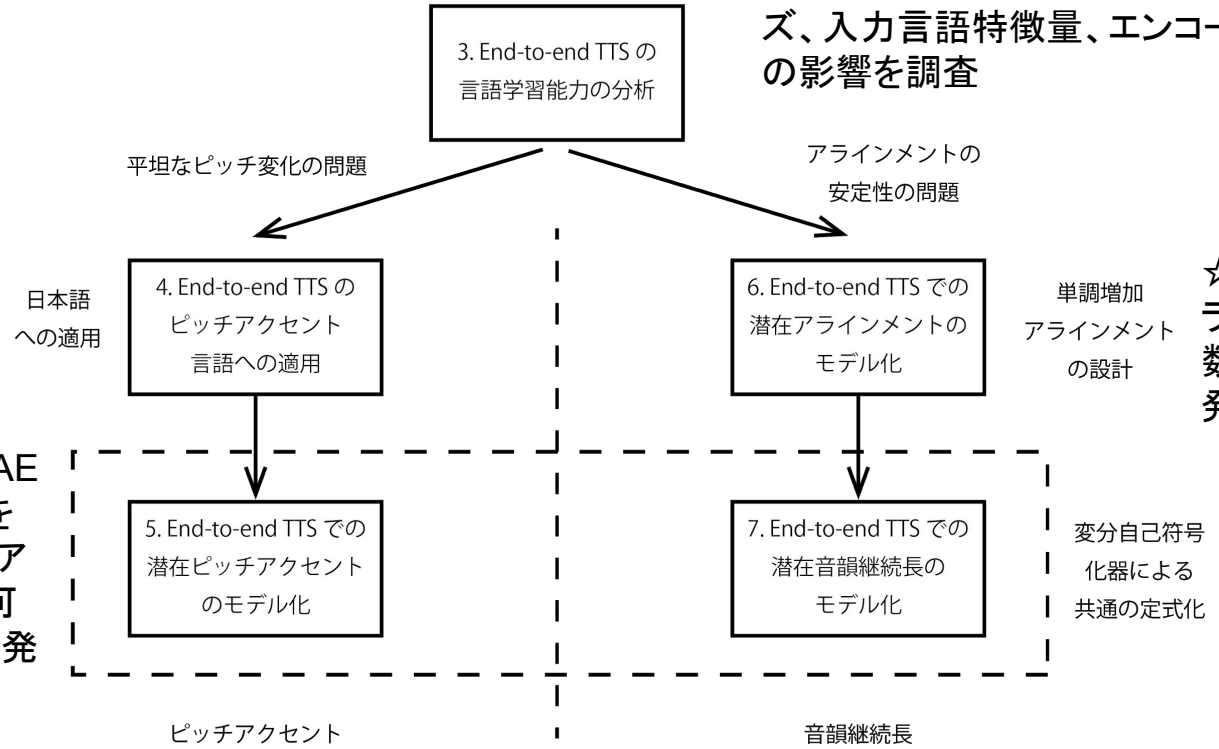
Method	Teacher	Student	Training phases	Aligner	Duration predictor
FastSpeech	AR-Transformer	FF-Transformer	3	Soft-attention	Conv+Linear
DurlAn	Tacotron-like	-	3	HMM?	LSTM
FastSpeech2	FF-Transformer	-	3	HMM-GMM	Conv+Linear
AlignTTS	FF-Transformer	-	4	SSNT-like MDN	FFT+Linear
JDI-T	AR-Transformer	FF-Transformer	1	Soft-attention + CTC	Conv+Linear
Glow-TTS	Glow	-	1	MAS	Conv+Linear
VQ-VAE	VQ-VAE	-	1	CTC	LSTM

- 問題点
  - 入力言語ラベルの設計に左右されやすい
    - ポーズの有無
    - 記号類など、継続長と対応関係がとりにくいもの
  - 強制アラインメントに用いる認識器（CTC）が継続長のサンプラーとして十分強力ではない
    - 条件独立性の仮定
    - 認識器ではなくセグメンテーションとして用いることの是非

8. 結論：まとめ

☆英語にてTacotronを用い、End-to-end法とパイプライン法を比較、パラメーターサイズ、入力言語特徴量、エンコーダーの構造の影響を調査

☆日本語にてTacotronを用い、アクセント型ラベルの有無、パラメーターサイズの影響を調査



☆単調増加な離散アラインメントを潜在変数に用いる手法を開発

☆条件付きVQ-VAEを用いて、ラベルを用いず潜在ピッチアクセントの予測を可能にする手法を開発

☆条件付きVQ-VAEを用いて、音素継続長を潜在変数とする手法を開発

8. 結論：解決した問題

音素継続長

- Soft-attentionを用いたアラインメントは不安定で、エラーを起こしやすい。
  - 離散アラインメントを用いる手法を提案。単調増加なアラインメントを設計でき、致命的なアラインメントエラーを回避できる。
- Soft-attentionを用いる現在のEnd-to-end法ではフレーム単位のアラインメントを用いているが、音声生成や計算効率の観点から、音韻継続長でアラインメントを表現したほうがよい。
  - 条件付きVQ-VAEを用いることで、継続長モデルと強制アラインメントを End-to-end TTSに組み込むことができる手法を提案。
  - 音韻継続長を潜在変数として用いることで、単調増加なアラインメントをより効率よく実現できる。

8. 結論：未解決の問題

- 英語の平坦なピッチの改善
  - 3章での英語のEnd-to-end TTSが示した平坦なピッチ問題は解決されていない。
  - 5章のピッチアクセントモデルを英語に導入し、英語のピッチアクセントの改善を行う。
- 日本語のピッチアクセントモデルの性能改善
  - 5章のピッチアクセントモデルは、ほどほどに正しいピッチアクセントしか達成しなかった。
  - ピッチアクセントは単語や句に紐づくので、音素からの予測では限界がある。
  - テキストを入力に用いる日本語 End-to-end TTSを実現して、その後ピッチアクセントモデルをそれに導入して性能が改善するか検証する。
- 離散アラインメントを用いた手法の自然性改善
  - 状態遷移、音素継続長をアラインメントに用いる手法いずれも合成音声の自然性が十分ではない。
  - 離散アラインメントの性能改善
    - 様々な形式、品質のラベルに対応
    - 継続長のサンプラーの性能改善

博士論文のもとになった論文

ジャーナル論文

- Yusuke Yasuda, Xin Wang, Junichi Yamagishi,“Investigation of learning abilities on linguistic features in sequence-to-sequence text-to-speech synthesis,” Computer Speech and Language, Volume 67, 2021, 101183 (2, 3, 4章)

国際会議論文

- Yusuke Yasuda, Xin Wang, Junichi Yamagishi, “End-to-End Text-to-Speech using Latent Duration based on VQ-VAE”. Proc. (ICASSP 2021に投稿, 7章)
- Y. Yasuda, X. Wang and J. Yamagishi, "Effect of Choice of Probability Distribution, Randomness, and Search Methods for Alignment Modeling in Sequence-to- Sequence Text-to-Speech Synthesis Using Hard Alignment," ICASSP 2020 - 2020 pp. 6724-6728 (6章)
- Y. Yasuda, X. Wang, J. Yamagishi, “Initial investigation of encoder-decoder end-to-end TTS using marginalization of monotonic hard alignments”. Proc. 10th ISCA Speech Synthesis Workshop, 211-216, 2019. (6章)
- Y. Yasuda, X. Wang, S. Takaki and J. Yamagishi, "Investigation of Enhanced Tacotron Text-to-speech Synthesis Systems with Self-attention for Pitch Accent Language," ICASSP 2019 pp. 6905-6909 (4章)

5章の内容は未投稿

公開したソースコード

- Self-attention Tacotron
  - https://github.com/nii-yamagishilab/self-attention-tacotron
- Tacotron2
  - https://github.com/nii-yamagishilab/tacotron2
- SSNT-TTS
  - TBD
- Conditional VQ-VAE
  - TBD