ポーズ統制と発話速度・対話速度の導出

吉村 有弘[1,a)]　幡手 千華[2]　浅井 紀久夫[3]

概要：音声合成（Text-to-Speech，TTS）では Speech Rate とポーズを設定し発話を合成できる．その利用は今後広がっていくと思われる．意図的にポーズが取られた発話の速度を分析する際，従来のようにポーズを含む条件で発話速度を導出する必要があるのだろうか．また，複数の長い音声ファイルの発話速度を導出するために，音声分析ソフトで発話区間とポーズを手動で見極めるのは時間を要する．そこで本発表では，librosa で 15〜20dB の音圧レベルでポーズを統制して発話速度を導出する方法を考察した．その結果， 多くの交替潜時を 0.2 秒未満に統制することができたが，短くカットされる発話があった．しかし，簡易に発話速度・対話速度を導出することができるのを確認した．

キーワード：モーラ, ポーズ，交替潜時，発話速度，対話速度，JLPT

1. はじめに

1.1 発話速度とは

発話速度は，流暢性，避難誘導，対話システム，日本語教育などの様々な分野で考察される指標である．日本語の発話速度は，仮名数を発話量とするよりも，モーラ数（拍数）を発話量として計算される傾向にある．

1.2 交替潜時とは

現話者から次話者へ発話権が移行する過程をターンテイキングと呼ぶ [1]. ターンテイキングの長さは交替潜時と呼ばれる．

1.3 スプレッドシートでのモーラ数の計算

モーラ数は，発話を仮名表記したものから，小さい拗音「ャ，ュ，ョ」を除した数である [2]．モーラ数は，表計算ソフトで計算するのが便利である．例えば，図 1 のようにスプレッドシートの B1 セルに片仮名入力した「キョウトトオオサカハカンサイデス」を B2 セルで仮名数を計算するには， 数式を= LEN（B1）とすることで 16 文字と計算ができる．B3 セルのように SUBSTITUTE 関数を使って小さい拗音「ャ，ュ，ョ」を除くことで，「キウトトオオサカハカンサイデス」とモーラ表記が可能である．さらに，LEN 関数を組み合わせれば，セル B4 のようにモーラ数を 15 と計算ができる．

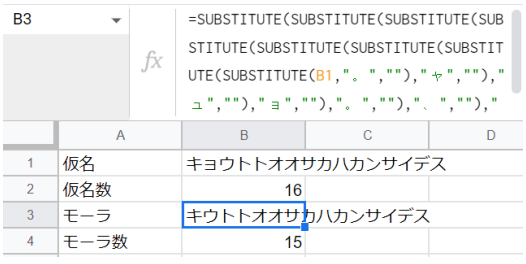

図 1　スプレッドシートでのモーラ数の計算

[1] 放送大学大学院
[2] 福岡女子大学大学院
[3] 放送大学
[a)] 2028001710 @ campus.ouj.ac.jp

2. Python での発話速度導出

2.1 モーラの表記

Python でコード 1 のようにすることで，音声認識した漢字仮名混じり表記をモーラ表記に変換可能である．

1) 音声認識機能 [3] を使って wav ファイルを漢字仮名混じり表記に文字起こしをする．
2) MeCab の Oyomi[4] で読みを片仮名変換する．
3) 小さい拗音「ャ，ュ，ョ」を除く．

```
onsei = "ファイルパス"
ninshiki = speech_recognition.Recognizer()
with speech_recognition.AudioFile(onsei) as
   source:
    audio = ninshiki.record(source)
kanji = ninshiki.recognize_google\
    (audio, language='ja-JP')
mecab = MeCab.Tagger("-Oyomi")
kana = mecab.parse(kanji)
         # 小さい拗音を削除
kana = kana.replace("ャ", "")
kana = kana.replace("ュ", "")
kana = kana.replace("ョ", "")
mora = kana.replace("␣", "") # 空白を削除
print(mora)
```

コード 1 wav ファイルからモーラを表記するコードの一部

2.2 発話速度の計算

Python でコード 2 のようにすることで，発話速度の計算が可能である

1) librosa[5] でオーディオ長を取得する．
2) Len でモーラ数を数える．
3) モーラ数をオーディオ長で除算する．

```
y,sr = librosa.load(onsei)
time = librosa.get_duration(y=y, sr=sr)
mps = (len(mora)-1)/time
print("モーラ表記：", mora)
print("モーラ数：", len(mora)-1)
print("長さ：", time)
print("発話速度：", mps)
```

コード 2 発話速度を計算するコードの一部

3. ポーズと休止区間

籠宮ら（2008）によれば，「ポーズ」とは，転記基本単位の切れ目となる，主に 200ms 以上のポーズである [6].

発話の分析対象は自然発話だけではなく，TTS で作られた合成音声も分析対象になりえる．

例えば，Google TTS，Amazon Polly TTS，Microsoft Azure TTS では，表 1 のように，SSML 機能を使ってポーズの設定が可能である．

表 1 主要 TTS の SSML でのポーズの設定

TTS 名	SSML でのポーズの設定
Google	`<break time="0.2s"/>`
Amazon Polly	`<break time="0.2s"/>`
Microsoft Azure	`<break time="0.2s"/>`

発話速度の計算において，TTS で設定したポーズをどの程度まで含めるかは基準がない．また，Praat などの音声分析ソフトを使って，複数の長い音声ファイルの発話部分を見極めるのは時間を要する．

そのため，本発表では便宜上文法の切れ目は考慮せず，0.2 秒以上の発話の休止区間をポーズ，0.2 秒未満の休止区間を単に休止区間として扱う．そして，librosa[5] を利用してポーズを統制することで，音声ファイルからポーズをなくし，発話速度を適切に導出できるかを検証する．

4. ポーズ統制と発話速度・対話速度導出

4.1 JLPT の交替潜時

『日本語能力試験公式問題集 第二集』[7] の課題理解では，男女が交替で発話する．その 210 の発話の交替潜時を Praat で確認した．その結果，ほとんどの話者交替は 0.2 秒以上であった．JLPT の交替潜時は自然にできたものではなく，意図して調整されていると考えられる．TTS で設定した交替潜時と似ていることから，本発表では，日本語能力試験（JLPT）の交替潜時を 0.2 秒未満に統制できるか考察する．

図 2 は，JLPT 課題理解の N1〜N5 の交替潜時を箱ひげ図で表したものである．箱部分の横線が

中央値，三角マークが平均値，点が外れ値を表す．N5 からレベルが上がるに連れて交替潜時が短く設定されていることがわかる．

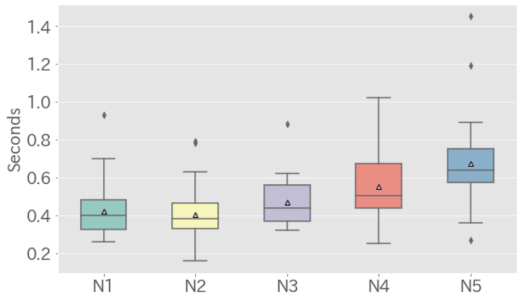

図 2　JLPT 課題理解の交替潜時

4.2　librosa でポーズを統制

コード 1 とコード 2 の間でコード 3（librosa[5]）のように音圧レベル 15 を閾値として，wav ファイルのポーズを統制する

```
#wavファイルのポーズを15dBを閾値でカット
clips = librosa.effects.split(y, top_db=15)
print(mora)
```

コード 3　wav ファイルのポーズをカットするコードの一部

コード 3 を 210 の wav ファイルに適用すると，多くの話者交替を 0.2 秒未満に統制できることを確認した．

しかしながら，卓球（たっきゅう）の促音，喫茶店（きっさてん）の「きっ」，「です」の「す」などのことばが短くなった．そのため，音圧レベルは 20db〜30dB 程度にした方がよい可能性がある．またポーズが厳密にどの程度短くなったかについて追加調査が必要である．

4.3　発話速度・対話速度の導出

通常発話速度は話者ひとりの発話速度を計測する．しかし本研究では，交替潜時が統制できたかを確認するために，2 人の話者が対話する速度を分けずに計測した．

『日本語能力試験公式問題集 第二集』[7] の課題理解のポーズを統制した後の対話速度は図 3 の N1L〜N5L である．ポーズ統制前後の対話速度は同じ順序であり，複数話者の発話の速度を概ね適切に計測できたと考えられる．

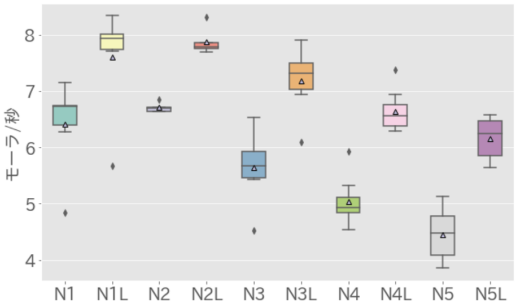

図 3　JLPT 課題理解のポーズ統制前後の対話速度

5.　まとめ

librosa[5] を利用することで，交替潜時の多くを 0.2 秒未満に統制することができるとわかった．しかしながら，一部の発話が途切れるため，精度を保って発話速度を計測するには，音圧レベルの調整が必要である．

本手法を利用することで，発話速度・対話速度の計測を簡易にできることがわかった．

参考文献

[1] 井上晃司, 河原 達也: **音声対話システム 基礎から実装まで**. オーム社, 2022.

[2] 飯塚元, 飯塚恵美子: **日本語の音声入門**. バベルプレス, 2018.

[3] SpeechRecognition: https://pypi.org/project/SpeechRecognition （2022 年 12 月 5 日アクセス）

[4] Taku Kudo: **MeCab: Yet Another Part-of-Speech and Morphological Analyzer**. https://taku910.github.io/mecab/ （2022 年 12 月 5 日アクセス）

[5] librosa: https://librosa.org/ （2022 年 12 月 5 日アクセス）

[6] 籠宮隆之, 山住賢司, 槙洋一, 前川喜久雄: **自然音声における大局的な発話速度の近くに影響を与える要因**. 音声研究, Vol.12, No.1, 2008.

[7] 日本語能力試験公式ウェブサイト: **日本語能力試験公式問題集 第二集**. 2018. https://www.jlpt.jp/samples/sampleindex.html （2022 年 12 月 5 日アクセス）