5P-05

誤分割された音声発話の修復手法のオンライン実装とユーザ適応

堀田 尚希 † 駒谷 和範 ‡ 佐藤 理史 † 中野 幹生 ‡‡

† 名古屋大学 大学院工学研究科 ‡ 大阪大学 産業科学研究所 ‡‡ ホンダ・リサーチ・インスティチュート・ジャパン

1. はじめに

人型ロボットやバーチャルヒューマンエージェントなど，擬人化されたシステムとの音声対話では，人間同士の対話と同様に，ユーザの発話に対して素早く応答することが望まれる．一方でシステムが素早く応答しようとする場合，システムがユーザ発話中の短い無音区間を発話終了であると誤って認定することがある．その結果，ユーザが発話中であるにもかかわらずシステムが話し始めてしまう問題が発生する．このときシステム内部では，元来一発話であったユーザ発話が，無音区間により複数の発話区間に分割されている．本研究ではこの現象を発話の誤分割と呼ぶ．また誤分割された発話断片をそれぞれ前半断片，後半断片とする．

本稿ではまず素早い応答を実現するため，この誤分割の修復をオンラインで行うシステムの実装を行った．我々はこれまでに，発話の誤分割により発生する，音声認識誤りと不適切なターンテイキングを，事後的に修復するシステムを MMDAgent[1] 上に実装してきた [2]．このシステムでは統合解釈時にシステム応答の遅延が生じていた．統合解釈のオンライン化を行うことで，統合解釈時にも素早い応答を実現する．

次に誤分割修復のユーザ適応を行うことで，修復が必要か否かの判定精度を向上させる．具体的にはユーザの発話テンポが個人により異なることに着目し，このテンポに応じて修復が必要か否かを判定するパラメータ (発話間間隔) を変更する．

2. 誤分割修復のオンライン実装

2.1 システムの概要

本章では統合解釈処理の高速化について説明する．従来のシステムでは，後半断片の発話が終了してから，前半断片と後半断片を結合して再度音声認識を行っていた．このため統合解釈処理により応答に遅延が生じていた．

我々は発話区間検出 (VAD) のパラメータを変えた 2 つの音声認識エンジン (Julius) を並列に動作させることで，統合解釈処理を仮想的にオンライン化する．並列に動作させる 2 つの Julius は音声区間終了部のマージン長 (-tailmargin) が異なっており，これらを Julius-Short，Julius-Long とする．-tailmargin の値は，それぞれ 0.24[秒]，2.00[秒] とする．前者は MMDAgent で用いられる Julius のデフォルト値であり,後者は-tailmargin を十分に長くした場合に相当する．つまり基本的に Julius-Short により素早い応答を実現し，発話の誤分割が生じている可能性が高い場合には，並列して動作させている Julius-Long により得られる結果を用いる．この 2 つの VAD 結果の例を図 1 に示す．どちらの Julius の VAD 結果を使用するかを決定することは，修復が必要か否かを判定することと同義である．これを 5 つの特徴を用いた決定木により判定する手法は，既に提案済みである [2].

Restoring Incorrectly-Segmented Spoken Utterances: Online Implementation and User Adaptation: *Naoki Hotta (Nagoya Univ.), Kazunori Komatani (Osaka Univ.), Satoshi Sato (Nagoya Univ.), and Mikio Nakano (Honda Research Institute Japan Co., Ltd.)*

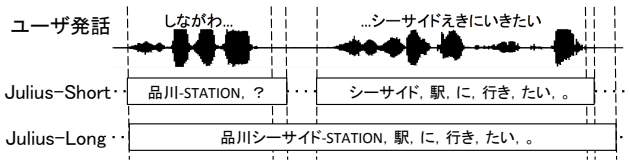

図 1: パラメータの異なる 2 つの VAD 結果の併用

表 1: 誤分割修復時におけるシステム応答の遅延時間

	平均 [秒]	標準偏差 [秒]
従来実装	2.84	1.11
提案実装	1.02	0.05

2.2 応答速度の実験的評価

提案実装が我々の従来の実装と比べてより素早く応答できることを，統合解釈時におけるシステム応答の遅延時間を比較することにより示す．ここではシステム応答の遅延時間を，ユーザの後半断片の発話終了時刻 (Julius-Short の VAD による発話終了時刻) と統合解釈を行った結果に基づくシステム応答開始時刻の差と定める．

修復が必要である可能性のある発話断片の対に対して，応答速度の評価を行った．ここでは我々が以前に構築したレストラン検索システムにより得られた発話断片の対のうち，(1) 発話間間隔が 0.90 秒未満，(2) 発話断片の長さが 0.80 秒以上，(3) 正解データにキーワード (地名，店名，駅名など) が含まれている，の条件を満たす，計 153 対を評価データとした．

提案実装でのシステムの応答の遅延時間を従来実装と比較した．従来実装は，後半断片が終了してから発話断片を結合し，再度音声認識を行うものである．結果を表 1 に示す．誤分割の修復を行う場合，従来の実装法では平均 2.84 秒の遅延が生じていたが，本稿で提案する実装法では遅延が平均 1.02 秒と，約 64%減少した．遅延時間が減少したのは，統合時の音声認識をオンラインで行っているためである．これに加え，実装したシステムでは，遅延時間の標準偏差が小さくなっている．これは，どのような発話断片対に対しても応答のための遅延時間が変わらないことを意味する．従来実装では，後半断片終了後に再度音声認識を行うため，結合する発話の長さに応じて遅延時間が変動していた．どのような発話断片対に対しても素早い応答ができるという点で，提案実装は従来の実装に比べ優れている．

3. 修復要否判定のユーザ適応

3.1 ユーザ適応処理の概要

発話の誤分割の修復が必要か否かの判定精度を向上させるため，ユーザのそれまでのふるまいに適応した判定を行う．この判定において，発話断片間の時間間隔に対する閾値は重要なパラメータである [2]．本章ではこの閾値をユーザに応じて変化させることにより，修復が必要か否かの判定精度の向上を目指す．

適切な発話間間隔の閾値は，ユーザの話すテンポと関係があると考える．ここではあるユーザの発話テンポを，当該ユーザの対話データにおける，システムの発話終了

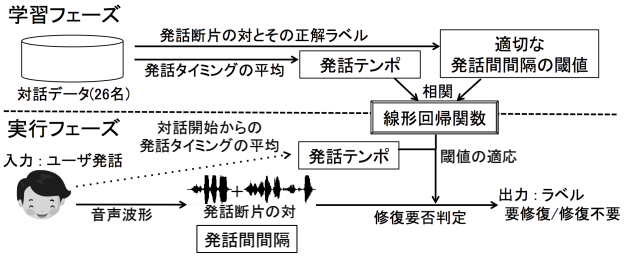

図 2: 発話テンポに基づく修復要否判定のユーザ適応

からユーザの発話開始までの時間間隔の平均とする．システムに対して素早く応答するユーザは，発話中の言い淀みが少なく，発話中のポーズも短いと予想される．このため発話間間隔の閾値を短く設定し，不要な修復処理やそれに伴う応答の遅延を避ける．一方で，システムにゆっくりと応答するユーザは，発話中に長いポーズがある可能性がある．この場合，発話間間隔の閾値を長く設定し，発話断片間の間隔が長い場合でも発話断片対を修復するのが望ましい．

ユーザの発話テンポは，各ユーザに対して個別に，対話開始から現在の発話までの発話タイミングの平均として計算する．本稿ではこれをオンライン適応と呼ぶ．これに対し，各ユーザの対話全体における発話テンポを用いる場合をバッチ適応と呼ぶ．バッチ適応は対象ユーザの対話データが全て事前に得られていると仮定した場合に相当し，理想的な条件である．

ユーザ適応の処理の流れを図 2 に示す．ユーザ適応は学習フェーズと実行フェーズの 2 つに分けられる．学習フェーズでは, ユーザ毎の対話データから, 適切な発話間間隔の閾値と発話テンポの相関を調べ, それらの線形回帰関数を求める．実行フェーズでは, ユーザの一発話を入力とし, それまでの対話履歴から得られる発話テンポと，学習フェーズで得た線形回帰関数を用いて，修復要否判定に用いられる発話間間隔に対する閾値を適応させる．その後，この適応した修復要否判定部により，修復が必要であるか否かのラベルを出力する．

3.2 ユーザの発話テンポと適切な発話間間隔の相関

ユーザの発話テンポと適切な発話間間隔の閾値の関係を調査した．本研究では我々が以前に構築した世界遺産検索システムにより収集されたデータのうち，対話ログが保存されている 26 名のユーザ，390 対の発話断片対を調査対象とした．ここでは発話間間隔が近接しており (2.00 秒未満)，雑音ではない (各発話断片が 0.80 秒以上) 発話断片対が 6 対以上あるユーザを調査対象とした．これらの発話断片対には，文献 [2] と同様に，元来一発話であるか否かのラベルを付与した．

適切な閾値は，データ中の発話断片対を用いて定めた．具体的には，ユーザ毎に，元来一発話か否かを，発話間間隔の特徴のみを用いて，Weka-3-6-9 の SMO により判定した場合の閾値とした．

これらの関係をプロットしたものを図 3 に示す．縦軸は適切な発話間間隔の閾値 [秒] を示し，横軸はシステムの発話終了からユーザの発話開始までの時間間隔の平均 [秒]，すなわち発話テンポを示す．このときの相関係数は 0.63 であった．これにより，発話テンポと適切な閾値には中程度の相関があることが示された．

3.3 線形回帰関数に基づく修復要否判定

ユーザの発話テンポと適切な発話間間隔の線形回帰関数を用いて，ユーザ毎に発話間間隔の閾値を決定し，修

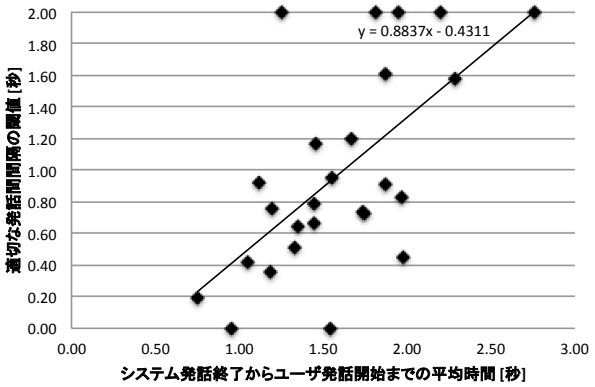

図 3: ユーザの発話テンポと適切な発話間間隔の相関

表 2: 閾値のみによる修復要否判定の精度

ベースライン	281/390 (72.1%)
オンライン適応	294/390 (75.4%)
バッチ適応	306/390 (78.5%)

復要否判定を行う．ここでは 3.2 節と同様に，26 名のユーザ，390 対の発話断片対を調査対象とした．

まず図 3 のデータをもとに，ユーザの発話テンポと適切な発話間間隔との線形回帰式として，式 1 を得た．

$$\text{閾値 [秒]} = 0.8837 * \text{発話テンポ [秒]} - 0.4311 \qquad (1)$$

この回帰式を用いて，図 2 に示すように，ユーザの発話テンポから，閾値を決定する．

ユーザに適応した閾値を基に，修復要否判定の性能を確認する実験を行った．ここでは入力された発話断片対の発話間間隔が，ユーザに適応した閾値よりも小さい場合は，修復が必要であるとし，大きい場合は不要であると判定する．結果を表 2 に示す．ここでベースラインは，全てのユーザに対して単一の閾値 (0.8224[秒]) を設定した場合の精度を示す．これは，全てのユーザのデータにおける，適切な閾値である．オンライン適応により閾値のユーザ適応を行った場合は，ベースラインに比べて判定精度が 3.3 ポイント上昇した．またバッチ適応により閾値を決定した場合は，ベースラインに比べて判定精度が 6.4 ポイント上昇した．このことから閾値のユーザ適応により判定精度が向上することが確認できた．オンライン適応では，対話の開始時付近での適応が不安定であることから，バッチ適応ほど性能は向上しなかったものの，ベースラインよりも性能が向上することが示された．

本稿では，単純な閾値処理において，ユーザ適応による修復要否判定の性能向上を示した．今後，様々な特徴を用いた決定木においても，ユーザ適応により判定性能が向上することを示す．また，修復要否判定のユーザ適応時の性能向上を，交差検定でも評価予定である．ただし，線形回帰式のパラメータ数は 2 個であることから，交差検定の場合でも同等の性能が得られると考えている．

参考文献

[1] A. Lee, et al., “MMDAagent - a fully open-source toolkit for voice interaction systems,” in *Proc. IEEE-ICASSP*, 2013, pp. 8382–8385.

[2] N. Hotta, et al., “Detecting Incorrectly-Segmented Utterances for Posteriori Restoration of Turn-Taking and ASR Results,” in *Proc. INTERSPEECH*, 2014, pp. 313–317.