Vol.2017-NL-231 No.2
Vol.2017-SLP-116 No.2
2017/5/15

応答義務推定の補助としての繰り返し発話検出

川井 雄太[1,a)]　藤田 寛泰[2,b)]　谷川 晃大[1,c)]　山下 峻[3]　船越 孝太郎[4,†1,d)]

概要：応答義務推定とは，発話区間に含まれる音響情報とユーザの振る舞いを特徴量とし，対話ロボットに応答すべきユーザ発話を推定する能力を与える技術である．しかし，ユーザの振る舞いは人それぞれであり，未学習の振る舞いに対して応答義務推定が失敗することは避けられない．この時発生するエラーの 1 つがユーザ発話の無視である．一方で，発話を無視されたユーザは同じ発話の繰り返しでエラーから回復を図ると期待される．そうであれば，高精度に繰り返し発話を検出することで，応答義務推定の失敗をカバーし，インタラクションの質を向上できると考えられる．本研究では応答義務推定の補助を目的とし，機械学習を用いた繰り返し発話検出を試みる．合成音声と人間音声を用いた評価実験により，提案特徴量の有効性を確認できた．

Repetition Detection as a Failure Recovery of Response Obligation Estimation

Kawai Yuta[1,a)]　Fujita Hiroyasu[2,b)]　Tanikawa Akihiro[1,c)]　Yamashita Shun[3]
Funakoshi Kotaro[4,†1,d)]

1. はじめに

公共の場で人間と音声対話可能なロボットの実現が期待されている．こうした場でロボットを利用するために，ロボットは 2 つの課題をクリアする必要がある．1 つ目は，ロボットに向けられたユーザ発話だけでなく，足音や周囲の音楽など，応答すべきでない音が入力されることである．ロボットはユーザ発話とその他を区別し，ユーザ発話にのみ応答しなければならない．2 つ目は，ユーザ発話の中に応答すべきでないものが存在することである．この課題はロボットが一度に複数のユーザと対話する状況で発生する．ロボットはユーザ発話だとしても，それが他のユーザへ向けられた発話ならば応答すべきでない．この 2 つの課題をクリアできなければ，ロボットは雑音やユーザ同士の会話など，応答すべきでない発話に誤応答してしまう．

こうした課題を解決するために，Sugiyama らにより応答義務推定技術が提案されている [1]．応答義務推定技術とは，入力された音に対し，応答義務があるか判別する技術である．

応答義務推定はユーザとロボットが対話した時に発生する全ての音を対象とし，各入力音の区間 (以降，入力音区間) を「応答義務あり」「応答義務なし」のどちらかに分類する．分類に際して特徴量として使用する情報は音と，入力音区間および入力音区間後の数秒間に含まれるユーザの動きから抽出する．

図 1 はロボットと複数のユーザが対話している場面である．この場面では，ロボットは自身に向けられた手前のユーザ発話にのみ応答し，ユーザ同士で対話する奥 2 人の発話を棄却することが期待され，応答義務推定によりこれが可能になる．

1 (株)Nextremer
Nextremer Co., Ltd.
2 高知工科大学
Kochi University of technology
3 北海道大学
Hokkaido University
4 (株) ホンダ・リサーチ・インスティチュート・ジャパン
Honda Research Institute Japan Co., Ltd.
†1 現在，京都大学
Presently with Kyoto Uniersity
a) yuta.kawai@nextremer.com
b) hiroyasu.fujita@nextremer.com
c) akihiro.tanikawa@nextremer.com
d) funakoshi@jp.honda-ri.com

Vol.2017-NL-231 No.2
Vol.2017-SLP-116 No.2
2017/5/15

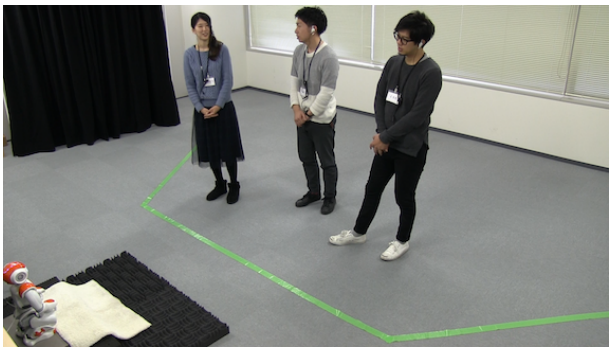

図 1 複数のユーザとロボットとの対話

しかしながら，ユーザの振る舞いは人それぞれであるため，全てのユーザの振る舞いを事前に学習しておくことは困難である．したがって，未学習のユーザの動作に対しては応答義務推定に失敗する確率が高くなる．

応答義務推定に失敗したロボットは，以下に示す 2 つのエラーのいずれかを起こすことになる．

(1) 応答の必要がない音に応答する
(2) 応答すべきユーザ発話を無視する

このうち，ロボットが応答すべきユーザ発話を無視した場合，ユーザは同じ発話を繰り返すことで，エラーから回復を図ると期待される．ロボットがユーザの繰り返し発話を検出し，本来応答すべき発話を無視していたと気づくことができれば，応答義務推定自体の精度はそのままでも，全体としてのインタラクションの質は向上すると予想される．したがって本研究では，応答義務推定の補助することを目的として，繰り返し発話検出を行なう．

2. 関連研究

システムが音声認識に失敗したことをユーザの繰り返し発話により検出し，再認識に役立てる研究は以前より行われている．これらの研究では，音声信号より直接得た特徴量と，音声認識結果から得た特徴量を利用する場合が多い．本研究ではそれぞれ，音声レベルの特徴量，文字レベルの特徴量とよぶ．

Kitaoka ら [2] と Levitan ら [3] は双方とも，誤認識した発話と直後の発話から上記のような特徴量を抽出し，機械学習による分類を行なう．本節ではこれら 2 つの研究の概要を述べる．

2.1 地名入力タスクにおける訂正発話検出

Kitaoka らはカーナビゲーションシステムで目的地を設定するタスクにおいて，システムが音声認識に失敗し誤った地名が入力された場合にユーザが発する訂正発話を検出する．Kitaoka らの手法では，最小累積距離 (音声レベル) と，音声認識候補集合の重なり度 (文字レベル) の 2 つの特徴量を用いる．

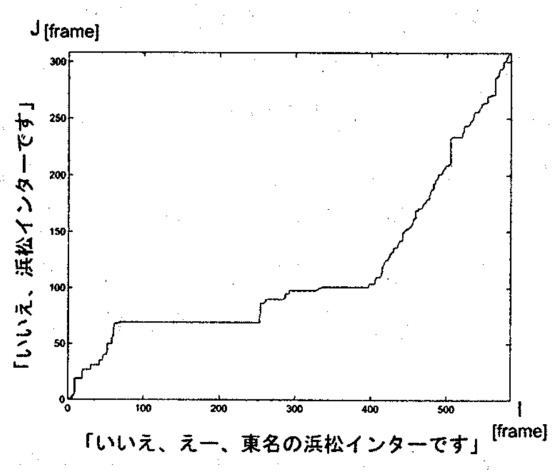

図 2 DP マッチング結果の例 出典：北岡ら (2003)[4]

2.1.1 最小累積距離 (音声レベルの特徴量)

最小累積距離とは，直前発話と現発話，2 発話の MFCC 系列に対し DP マッチングを行い，その結果得られる DP パスの照合終了点における累積距離である．DP マッチングは系列マッチング手法の 1 つであり，長さの異なる 2 系列の類似度を図るのに適している．

図 2 は連続する 2 発話間の DP マッチングを行った結果の例である．この DP マッチングでは，縦軸を直前発話のフレーム長，横軸を現発話のフレーム長とするトレリス状の空間 (以下，DP トレリス) を考える．DP トレリスの各点に局所距離として各フレームでの MFCC ベクトル間のユークリッド距離を設定する．照合開始位置から照合終了点まで，累積距離が最小となるように移動するパスが DP パスである．図では，「いいえ，浜松インターです」と「いいえ，えー，東名の浜松インターです」で，音声の MFCC が類似する 2 部分 (「いいえ」と「浜松インターです」) において，DP パスが斜めに進んでいる．Kitaoka らの手法では，一定ステップ以上斜めに移動する区間を，類似度が高い，繰り返し候補区間として抽出する (以下，共通部分とよぶ)．優先して斜めに進むよう，斜めに移動するコストはやや小さめに設定されているため，斜めに移動する割合が大きいほど，パス終端での累積距離は小さくなる．

2.1.2 認識候補集合の重なり度 (文字レベルの特徴量)

重なり度とは，直前発話と現発話の音声認識候補の集合間における同一の単語が含まれる度合いである．重なり度は次の式で定義される．

$$\text{重なり度} = \frac{\sum_{k=1}^{K} 2 * \sqrt{C_k^A * C_k^B}}{N_A + N_B} \quad (1)$$

ここで C_k^A, C_k^B は，それぞれ発話 A, B 中で k 番目に比較対象となった単語の頻度を表し (比較対象となった単語の総数は K)， N_A, N_B はそれぞれ発話 A, B の中で DP パスより抽出された区間の音声認識候補数を表す．認識候補集合で要素が重複しているほど重なり度は大きくなり，完

全に一致していれば最大値 1 を取る．

2.1.3 ロジスティック関数を用いた確率推定

Kitaoka らの地名入力タスクにおける訂正発話検出では，全体を言い直す発話および部分的に共通部分を持つ発話，若しくは言い換えを含む言い直し発話を正例とし，それ以外の発話を負例としている．

そして各発話の共通部分に対し，最小累積距離と重なり度を計算する．それらを説明変数として，ロジスティック関数により，各発話が繰り返しである確率を算出する．さらに，確率値に対して人手で閾値を設け，閾値を超えた発話を訂正発話とする．本稿では，最小累積距離と重なり度の 2 変数をベースライン特徴量として用いる．

2.2 音声検索クエリにおける再試行検出

Levitan らは音声検索クエリ発話が再試行かどうか，さらにどういった再試行かを 4 種類に分類する．

(1) `NO RETRY` ：再試行でない
(2) `REPETITION` ：クエリの訂正を意図した単純な繰り返し
(3) `REPHRASE` ：クエリの訂正を意図した言い換えを含む繰り返し
(4) `SEARCH RETRY` ：同じクエリでの再試行を意図している
(5) `OTHER` ：クエリに共通する部分が含まれるが，訂正をする意図はない

OTHER に該当する例として，下の文では Weather in は 2 クエリで共通しているが，訂正を目的としてでなく，別エリアの天気を検索している．

- `Q1.Weather in New York.`
- `Q2.Weather in Los Angeles.`

Levitan らは発話タイプの分類に用いる特徴量として，次の 3 カテゴリを定義している．

(1) `similarity` ：類似性に関する特徴量
(2) `correctness` ：正確性に関する特徴量
(3) `recognizability` ：認識可能性に関する特徴量

similality に属する特徴量は，再試行では同様のクエリが繰り返された可能性が高いため，2 クエリ間の類似性を測定する．特徴量は，2 クエリ間の編集距離 (未加工値・正規化値)，2 クエリ間の共通単語数，最長共通単語列長 (相対・絶対) である．

correctness に属する特徴量は，最初のクエリが正しく認識され，ユーザがシステムの提示結果と対話を行ったか否かを表す．まず，第一認識候補の confidence スコアを，システム自身のパフォーマンスに対する見解として特徴量とする．さらに，confidence の確度を高めるため，ユーザからの信号を用いる．つまり，最初のクエリで提示された結果 (構造化もしくは非構造化) とユーザが対話したかどうかのブール値を特徴量とする．再試行では最初のクエリよりも，ゆっくりと発話が繰り返されることもあるため，母音の数をクエリの秒数で除算し，特徴量として利用する．

recognizability に属する特徴量では，誤認識される可能性が高いクエリの特性をモデル化することを試みる．誤認識された可能性が高いクエリは言語モデル (以下，LM) スコアが低く，代わりとなる認識候補数 (以下，代替発音量) が多いとの仮説のもと，最初のクエリの LM スコアと代替発音量を特徴量とする．さらに，クエリの長さと，再試行される可能性には相関があるとの仮説から，各クエリの文字数，各クエリの単語数，各クエリの秒数を特徴量とする．また，ほぼ同一だが再試行ではない OTHER に属するクエリを識別するため，各クエリに含まれる大文字の単語数，各クエリが質問語 (who, what, etc.) で始まるか否かを特徴量とする．

これら 3 カテゴリの特徴量をロジスティック回帰の特徴量とし，音声クエリを NO RETRY，REPETITION，REPHRASE，SEARCH RETRY，OTHER の 5 つの再試行タイプに分類する．本研究では，このうち NO RETRY(繰り返しなし) と REPETITION(同一発話の繰り返し) に限定して識別を行なう．また，単純な繰り返しにのみ焦点を当てているため，similarity カテゴリから音素列の編集距離を特徴量として採用する．

3. 提案手法

前章で述べた通り，我々は Kitaoka らが提案する，重なり度と最小累積距離をベースライン特徴量とする．そこに音声レベルの新規特徴量として DP パスにおける移動数に関する特徴量を追加する．また文字レベルの特徴量として Levitan らの研究より，音素列間の編集距離を導入する．

音声分析は HTK 3.4.1[*1]を用いて，表 1 に記す条件で行なう． 音声認識には Julius Japanese Dictation-kit v4.3.1[*2]を用いる．

表 1 音声分析条件

音声特徴量	12 次元 MFCC
サンプリング周波数	16kHz
分析窓	ハミング窓
フレーム幅	25ms
フレームシフト	10ms

3.1 DP パスにおける移動数

DP パスにおける移動数とは，DP パス上の縦・横・斜め 3 方向にそれぞれ移動したステップ数を指す．図 3 は横方向への移動数を図示したものである．本研究では，これら 3 方向に対して総移動数，平均連続移動数，最大連続移動数を求めて特徴量とする．

[*1] http://htk.eng.cam.ac.uk/
[*2] https://github.com/julius-speech/dictation-kit

Vol.2017-NL-231 No.2
Vol.2017-SLP-116 No.2
2017/5/15

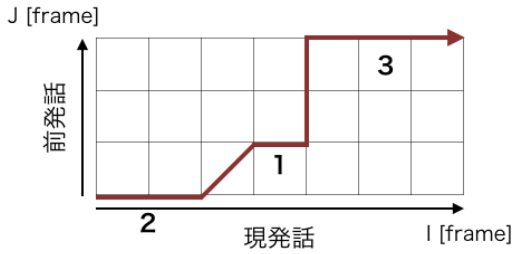

図 3　移動数の例

3.1.1　総移動数

総移動数とは，移動数の合計を指す．図 3 では横方向移動数に該当するのは 2，1，3 であるため，横方向の総移動数は $2+1+3=6$ となる．

3.1.2　最大連続移動数

連続移動数とは，同じ方向に連続して 2 以上進んだステップ数である．図 3 であれば，2，3 が横方向における連続移動数に該当する．最大連続移動数とは，各方向における連続移動数のうち，最大のものを指す．図 3 においては，最大連続移動数は $max(2,3)=3$ となる．

3.1.3　平均連続移動数

平均連続移動数は，各方向での連続移動数の平均値である．横方向平均連続移動数は，図 3 では $(2+3)/2=2.5$ となる．

3.2　編集距離

文字レベルの特徴量として，Levitan らの研究より編集距離を用いる．これは事前実験において，重なり度単体よりも，編集距離単体を特徴量とした方が 5-10 ポイント高い正解率を示したためである．Julius による音声認識 (DNN-HMM) を行い，10-best まで求めた音声認識候補に対して音素列を用いた編集距離を計算する．なお，表層文字列およびカナ文字列 (読み) よりも音素列に対して編集距離を求めたほうが正解率が高かった．

式 (2) に最小編集距離の定義を示す．

$$\text{最小編集距離} = \min_{1\leq i,j\leq N} ed(pre_i, cur_j) \qquad (2)$$

本研究では，編集距離は N-best 認識候補集合中，全要素を照合したうちの最小となる編集距離を用いる．pre_i は直前発話の i 番目の認識候補，cur_j は現発話の j 番目の認識候補，$ed(pre_i, cur_j)$ は pre_i と cur_j の編集距離，N は認識候補数数を表す．

3.3　先行研究との相違点

本研究とベースラインとする Kitaoka らの先行研究とでは，特徴量を抽出する区間や，特徴量分析条件が異なる点が存在する．表 2 はその相違点をまとめた表である．

Kitaoka らは間投詞など編集表現込みの訂正発話を扱うことから，発話区間から切り出した斜め移動区間に対してのみ特徴量を抽出していた．しかし，我々は単純な繰り返し発話のみを扱うため，発話区間全体から特徴量を抽出する．特徴量は重なり度と最小累積距離をベースライン特徴量とし，新たに 3 方向の移動数に関する特徴量，と編集距離を加えることを提案手法とする．機械学習アルゴリズムは，実験した範囲で最も正解率が良かった Random Forest を用いる．MFCC は HTK のデフォルト値である 12 次元で算出する．最小累積距離は DP トレリス中の最大累積距離で正規化する．文字レベルに関する特徴量は，Julius による音声認識結果を元に計算する．なお，Julius による音声認識は全てデフォルトのパラメータで行なう．認識候補解の数も，デフォルト値の 10 を用いる．

表 2　先行研究との相違点

	先行研究 [2]	本研究
検出対象	編集表現込みの訂正発話	単純な繰り返し発話
特徴量の抽出区間	斜め移動区間のみ	発話区間全体
特徴量	重なり度・最小累積距離	重なり度・最小累積距離・編集距離・移動数
機械学習手法	ロジスティック回帰	Random Forest
MFCC	10 次元	12 次元
最小累積距離の正規化	移動数で除算	最大累積距離で除算
音声認識エンジン	SPOJUS-SYNO	Julius(DNN)
N-best	200	10
語彙特徴量の元	認識結果の表層形	認識結果の音素列

4.　データ

実験では発話のペアから特徴量を抽出し，ペアが繰り返しか否かを判定する．本稿では合成音声を用いてモデルを学習する．音声合成を用いることで，大量のデータを簡単に確保できるためである．評価は合成音声による 10 分割交差検証と，人間音声をテストデータとする評価実験の 2 つにより行なう．

本節では合成音声の作成方法と人間音声の収録方法，合成音声・人間音声からどのようにデータセットを作成するかを説明する．

4.1　合成音声の作成方法

人間が同じ単語を繰り返し発声したとしても，毎回全く同じ音声信号となることはない．しかし，合成音声は工夫を施さなければ，毎回同じ信号で作成される．全く同じ信号でマッチングを行なうことは，現実的な問題設定ではない．そこで，同じ単語を発生する場合でも，モーラ間に記号を挿入してイントネーションを変更したり，話速を調整することで，同じ信号が生成されることがないようにする．また，公共の場で収録したノイズなどが含まれるユーザ発

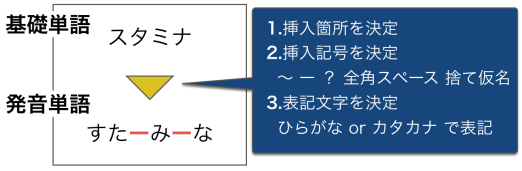

図 4　発音単語の作成

話に近づけるため，作成した合成音声には信号レベルで加工を施す．

4.1.1　音声合成

合成音声は Mac OS X 付属の say コマンドにより作成する．

say コマンドで作成した音声は，サンプリングレート 16kHz，ビット数 16bit，チャンネル数 1 のリニア PCM で保存する．

合成音声は以下に示す 4 ステップで作成する．

(1) 基礎単語の選択

基礎単語として，MeCab 0.996[*3]の名詞辞書から 4 モーラの単語を選択する． 単語を 4 モーラに限定するのは，フレーム数の差のみで繰り返し判定をすることを防ぐためである．本稿の問題設定では，単語の文字数が異なる場合，容易に識別できるケースが大半を占めてしまうため，トリビアルに識別ができないケースに限定する．

(2) 発音単語への変換

基礎単語をそのまま say コマンドで発声させても，毎回全く同じ信号が生成される．そのため，モーラ間に記号を挿入し，表記をひらがな・カタカナでランダムに選択することで発声時のイントネーションを変化させる．

図 4 は，基礎単語を発音単語に変換する例である．まずモーラ間のどの位置に記号を挿入するか決定する．挿入する記号は，長音，小書き文字，波線，全角スペース，疑問符の 5 つからランダムに選択する．1 つの発音単語中に 2 種類の記号が出現することはないようにする．最後にひらがな・カタカナ，いずれで表記するかを決定する．say コマンドでは表記文字種によってもイントネーションの変化が生じる．

(3) 話速の調整

話速は 150 モーラ/分～250 モーラ/分の範囲で設定する．これは say コマンドの rate オプションに値を設定することで行なう．

(4) say コマンドによる音声合成

(1)～(3) で設定した内容を元に say コマンドで合成音声を作成する．ここで作成した合成音声は，声の信号のみが綺麗に保存された音声なため，次項で説明する音声加工により現実での収録に近づける．

*3　http://taku910.github.io/mecab/

4.1.2　音声加工

公共の場で録音した音声は，環境音がノイズとして含まれるほか，発話区間が上手く切り出せず，録音途中で音声が途切れたり，逆に余計な空白を録音して冗長になることがある．このような環境を SoX[*4]による音声加工で再現し，現実的な問題設定に近づける．

音声は次の 2 ステップで加工する．

(1) 音声先頭・末尾のフレーム処理

SoX により，合成音声の先頭・末尾に無音区間を追加，もしくはランダムなフレーム数削除を行う．各音声，先頭，末尾と 2 箇所に処理を施す．施す処理は無音区間の追加，フレーム削除のいずれかである．追加する無音区間は 0～200ms の範囲でランダムに選択する．削除するフレーム数は 0～100ms の範囲でランダムに選択する．

(2) ノイズ重畳

フレーム処理後，SoX により合成音声と同じ長さのホワイトノイズを都度作成し，合成音声全体に重畳する．なお，本稿では合成音声とホワイトノイズの SN 比は 10.4 とし，大きめのノイズを重畳した．

4.2　人間音声の収録

ここでは，人間音声の収録方法を述べる．録音は MacBook Air 内蔵マイクで行なった．収録環境は作業音や話し声など，環境音が発話に混ざるように平日のオフィスで行った．これは公共の場における録音環境に近づけるためである．発話区間は Julius Japanese Dictation-kit v4.3.1 付属の adintool で自動的に切り出した．語彙は合成音声と同じく MeCab 名詞辞書より 4 モーラの単語を選択した．また，発音は以下の指示を与え，1 単語あたり 5 パターン収録した．

(1) 普通
(2) 普通
(3) 遅く
(4) 速く
(5) 強く

4.3　データセットの作成方法

本研究におけるデータセットとは，2 発話が同じ基礎単語を発声していれば正例，異なる基礎単語であれば負例とする発話ペアの集合である．合成音声・人間音声ともに同様の方法でデータセットを作成する．各話者毎に，全ての発話集合から，以下の条件を守るように発話ペアを取り出す．

(1) 基礎単語が同じなら正例，異なれば負例
(2) 正例，負例は 1:1
(3) 同じ発話ペアを取り出すのは一度だけ

*4　http://sox.sourceforge.net/

表 3　実験条件

	話者	語彙	発音	話速	音声合計
合成音声	日本語男性	44 単語	5 種	5 種	1100
人間音声	男性 2 名	26 単語	5 種		130

表 4　手法間における正解率の比較

	ベース	ベース+編集	ベース+提案	ベース+編集+提案
合成音声	70.2	74.3	85.1	87.2
人間音声	78.4	81.6	85.9	89.9

表 5　話者別での人間音声の正解率

	ベース	ベース+編集	ベース+提案	ベース+編集+提案
人間音声 A	82.8	88.4	90.0	96.2
人間音声 B	74.0	74.8	81.8	83.6

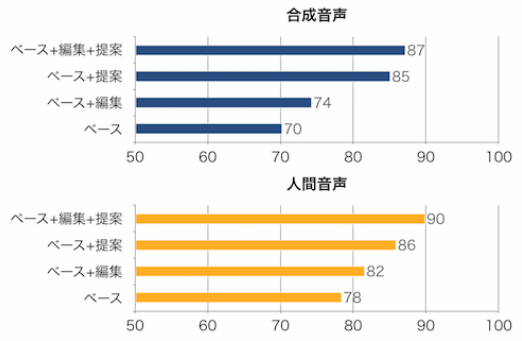

図 5　手法間における正解率の比較

5. 実験

本章では実験について述べる．実験は合成音声による 10 分割交差検証と，合成音声を学習データとし人間音声をテストデータとする，2 つの検証方法で行なう．

5.1 実験設定

実験条件を表 3 に示す．

合成音声は 44 語彙，発音は各基礎単語 5 パターン，話速は各発音単語 5 パターンで計 1100 作成した．合成音声によるデータセットはここから 26400 ペア作成した．人間音声は男性話者 2 名で 26 語彙，発音は各基礎単語 5 パターンで各話者 計 130 作成した．人間音声によるデータセットは話者毎に 500 ペア作成した．

5.2 実験結果

実験結果を表 4，図 5 に示す．なお，人間音声は表 5 に各話者毎の正解率を示すが，これ以降は 2 話者の正解率の平均値を基に議論を進める．

ベースライン特徴量 (ベースと表記) は最小累積距離と重なり度を指す．編集距離 (編集と表記) は最小編集距離を指す．提案特徴量 (提案と表記) は 3 方向の総移動数・平均連続移動数・最大連続移動数を指す．

表 6　移動数別の実験結果

	ベース+総移動数	ベース+平均連続移動数	ベース+最大連続移動数
合成音声	79.8	82.1	81.8
人間音声	83.0	85.8	85.7

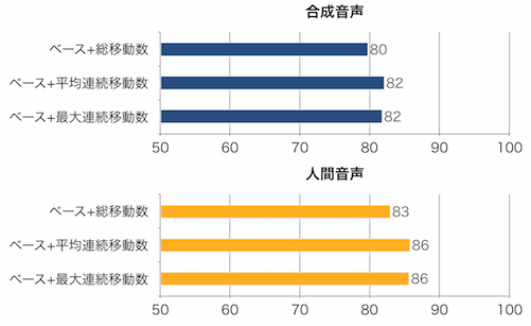

図 6　移動数別の実験結果

表 7　個別の特徴量による実験結果

	提案特徴量	最小累積距離	編集距離	重なり度
合成音声	82.8	67.9	72.8	67.0
人間音声	78.1	66.9	80.8	74.9

合成音声による 10 分割交差検証では，ベースライン特徴量に提案特徴量を付加することで正解率が向上した．また，人間音声による評価でも同様に正解率が向上した．このことから，単純な繰り返し発話検出を行なうタスクでは，合成音声で学習したモデルでも人間の繰り返し発話の検出に有効であることが示唆された．

5.3 実験の考察

提案手法についてより詳細に考察を行なう．ベースラインに総移動数，平均連続移動数，最大連続移動数をそれぞれ個別で加えた場合の精度を，表 6, 図 6 に示す．

結果は，ベースライン+総移動数でやや低く，ベースライン+平均連続移動数，ベースライン+最大連続移動数で総移動数よりも高い正解率となった．ここから，DP パスの移動数の平均値，最大値など，隠れた情報を上手く捉えることで，単に類似度 (最小累積距離) を特徴量とするよりも精度が向上する可能性が示唆されている．

表 7，図 7 に示すように，音声レベル，文字レベルの特徴量でみると，重なり度，編集距離といった文字レベルの特徴量が人間音声の正解率に寄与する割合が高い．一方，合成音声では最小累積距離，移動数といった音声レベルの特徴量が正解率に寄与する割合が高い．

これは，合成音声に大きめのノイズを重畳し，音声認識率が下がったためだと思われる．実際に，合成音声の音声認識率 (単語正解率) は 22%なのに対し，人間音声では 58%とそれよりも高い正解率で音声認識が出来ていた．文

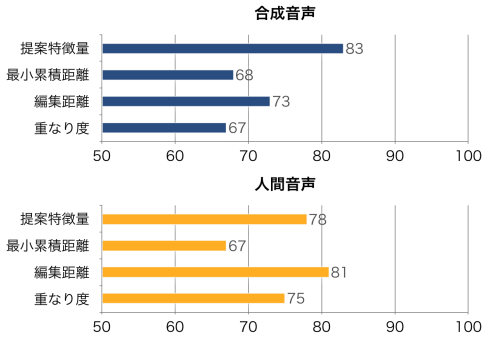

図 7 個別の特徴量による実験結果

字レベルの特徴量が音声レベルの特徴量よりも高精度となるとの知見もある [5]．しかし，合成音声の繰り返し検出に音声レベルの特徴量が有効であったように，音声認識が難しい環境下において音声レベルの特徴量が有効である可能性が示唆されている．これらのことから，音声レベルでの隠れた特徴量をより上手く抽出できれば，さらに精度が向上すると思われる．

5.4 今後の課題

今回は単純な繰り返し発話を対象に実験を行ったが，対話ロボットのタスクによっては，間投詞が含まれる繰り返しや，同じ内容を言い換えて発話し，対処することもある．今後はこうした編集表現が異なる場合であっても，一致できるように実験が対象とする発話内容の幅を拡大する必要がある．4 モーラに固定しない実験についても考える．

また，考察のように，DP パスからより有用な音声レベルの情報を抽出できれば，更に精度が向上する可能性があると考えている．このことから，RNN や HCRF といった，系列データをより直接的に扱えるアルゴリズムの適用で精度が向上する可能性が考えられる．

文字レベルの情報に関する予備実験では，認識結果の表層文字列よりも音素列を用いるほうが良い結果が得られた．音素列よりも更に下位の抽象表現として senone[6], [7] がある．senone の利用も検討したい．

6. まとめ

本稿では合成音声を訓練データとして，繰り返し発話の検出実験を行った．合成音声の発話ペアが繰り返しか否かを学習したモデルを用いることで，人間音声の繰り返し発話を高い正解率で検出することが出来た．このことから，本稿で対象とした単純な繰り返し発話を検出するタスクにおいては，合成音声で学習したモデルが人間音声の繰り返し発話検出に有効である可能性が示唆されている．

また，ベースライン特徴量に新しく特徴量を加えることで繰り返し検出精度の向上が確認された．単に DP パスにおける類似度 (最小累積距離) を特徴量とするのでなく，移動数の平均値や最大値など，隠れた特徴量を抽出することで精度が向上した．今後はより上手く音声レベルの情報を活用する学習アルゴリズムを利用することや，間投詞など編集表現込みの繰り返し発話に実験範囲を拡大していくことが課題として考えられる．

謝辞

本研究を進めるにあたり，名古屋工業大学大学院工学研究科 李研究室 李晃伸教授より，研究の進展に有用なアドバイスを頂きました．この場を借りて，深く感謝の意を示します．

参考文献

[1] Sugiyama Takaaki, Funakoshi Kotaro, Nakano Mikio, Komatani Kazunori: “Estimating Response Obligation in Multi-Party Human-Robot Dialogues”, In Proc. Humanoids 2015, pp.166-172, 2015.

[2] Norihide Kitaoka, Naoko Kakutani, Seiichi Nakagawa: “Detection and Recognition of Correction Utterance in Spontaneously Spoken Dialog”, In Proc. EUROSPEECH 2003, pp.625-628, 2003.

[3] Rivka Levitan, David Elson: “Detecting Retries of Voice Search Queries”, In Proc. ACL 2014, pp.230-235, 2014.

[4] 北岡 教英, 角谷 直子, 中川 聖一: “対話音声中の言い直し発話の検出と認識”, 情報処理学会研究報告, 2003-SLP-46, pp.31-36, 2003.

[5] 岡 隆一, 西村 拓一, 張 建新, 伊原 正典: “フレーム特徴の音素記号化に基づく語彙に依存しない音声検索”, 電子情報通信学会論文誌, Vol.J86, No.6, pp.764-775, 2003.

[6] M. Hwang, X. Huang: “Shared-distribution hidden Markov models for speech recognition”, IEEE Trans. Speech Audio Process, Vol.1, No.4, pp.414-420,Jan, 1993.

[7] George E. Dahl,Dong Yu, Li Deng,Alex Acero: “Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition”, IEEE Trans. Speech Audio Process,pp.30-42,2012.