Vol.2017-NL-231 No.3
Vol.2017-SLP-116 No.3
2017/5/15

LSTMによる繰り返し発話検出の高精度化

山上 諒太[1,a)]　船越 孝太郎[2,1,†1,b)]　菅野 重樹[1]

概要：現在，音声対話エージェントに対して注目が高まっている．しかし，今日の音声認識は精度が高くなっているものの，発話内容の誤認識は避けられないのが現状である．一方で, 一般的にユーザは誤認識が発生した際に, 誤認識された部分を言い直す, すなわち同じ発話を繰り返すといわれている．本稿では，LSTM を用いたモデルによる繰り返し発話の検出を 2 種類提案し，合成音声を用いて評価を行った．また，従来手法について,DP マッチングの結果を用いる方法と, 音声認識結果の文字列を比較する手法で評価を行い, 提案した手法と比較した．結果として，提案手法の方が従来手法よりも繰り返し発話検出の性能が高くなることを確認した．

Detection of Repetitions in Conversation with LSTM

Yamagami Ryota[1,a)]　Funakoshi Kotaro[2,1,†1,b)]　Sugano Shigeki[1]

1. はじめに

近年, Apple の iPhone に搭載されている Siri[1] や, NTT ドコモのスマートフォンに搭載されているしゃべってコンシェル [2] など，音声対話システムに対して注目が高まっている．この背景として，深層学習による音声認識の著しい高精度化がある．例えば，当時 30%以上の誤差が出てしまうほど困難な課題であった電話会話音声の認識において，深層学習を導入した結果誤差が 20%未満になった [3].

しかし，現在の音声認識部分においてもノイズに弱い点もある [4]．また，辞書に登録されていない単語，すなわち未知語が入力された場合，基本的にシステム側は登録されている単語またはその組み合わせとして認識を行う．したがって，システムは未知語を認識することが出来ない．社会では常に新しい表現や言葉が作られ続けるため，この未知語の課題は不可避である．

一方で，ユーザはシステムが発話を誤認識したと考えた際，同様の発話を繰り返し発話する傾向がある [5]．つまり，このような繰り返し発話を検出することが出来れば，システムが発話の誤認識を検出する手がかりの 1 つとすることが出来る．

発話が類似していたかどうかを判定する研究として，矢野らはカーナビの地名入力タスクにおける訂正発話の検出を行っている [6]．この研究では判定に DP マッチングの結果と音声認識結果の重なり度を用いる．また，Levitan らは音声検索クエリの再試行の検出を行っている [7]．これには，単語や文字レベルでの編集距離等の類似性カテゴリ，システムの出力に対してユーザが対話を行ったか等の正確性カテゴリ，クエリの長さ等の認識性カテゴリの計 3 カテゴリの特徴量を抽出し，ロジスティック回帰の特徴量とすることで判定を行っている．

本稿では，単語のみの発話を想定し，2 つの単語が同じ (繰り返し発話である) か異なる (繰り返し発話でない) かを判定することを課題とする．

前述の通り，現在の音声認識でもノイズに弱い点もあるのが現状である．また，近年ディープニューラルネットワークを用いた機械学習が分野を問わずに活用され，成果を出している．これは，ディープニューラルネットワークが入力情報から適切な特徴量を自動的に抽出，分析を行うためである．

[1] 早稲田大学
Waseda University
[2] （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 Kyotaro University
[a)] r_yamagami@sugano.mech.waseda.ac.jp
[b)] funakoshi@jp.honda-ri.com

したがって，今回繰り返し発話の検出における提案手法として，音声認識結果の文字列ではなく，音声波形の類似性を判定の基準とし，時系列データを扱うことが出来るニューラルネットワークである LSTM (Long Short-Term Memory) を使用する．

2. 繰り返し発話の作成

2.1 音声データの作成

発話音声のデータとして，川井ら [8] が作成した合成音声にひらがな 4 文字の単語を発話させたものを使用した．

川井らは異なる音で始まる 44 の単語を選び，以下のパラメータを変化させながら，1 種類の単語につき 25 個，すなわち計 1100 個の音声データを既存の音声合成ツールを用いて作成した．

- 与える文字列について，
  - ひらがなまたはカタカナ
  - 長音を付加
  - 小母音を付加
  - 全角空白を付加
  - 疑問符を付加
- 発声レートの変更

パラメータについて，与える文字列を変化させることで発音を変化させることが出来る．また，発声レートを変更することで発話の速さを変化させることが可能になる．

このようなパラメータを変化させて音声データを合成した後，発話区間検出のぶれを想定して音声データに対して前後に無音区間を付加，あるいはトリムを行った．その後，ノイズとしてデータ全体に対しホワイトノイズの付加を行った．

2.2 データインスタンス（発話ペア）の作成

以上のように作成した 1100 個の音声データに対して，2 つ 1 組となるように全組み合わせをとり，同じ単語の組み合わせの場合繰り返し発話として正例，異なる単語の組み合わせの場合は繰り返し発話ではないとして負例とラベルをつけ，インスタンスを作成した．その結果，正例のインスタンスは 13,200 個，負例のインスタンスは 591,250 個，合計 604,450 個のインスタンスを得た．

表 1 インスタンスの内訳
Table 1 Type of Instance

正例 (繰り返し発話である)	負例 (繰り返し発話でない)	合計
13,200	59,1250	604,450

3. 従来手法

3.1 DP マッチングの結果による判定

DP マッチングは，長さが異なる 2 つの時系列パターンの類似度を求める手法である．

2 つの時系列パターン $A = a_1, a_2, \cdots, a_i, \cdots, a_I$ および $B = b_1, b_2, \cdots, b_j, \cdots, b_J$ について考える．ただし a_i および b_j は特徴量ベクトルである．

まず，i 行 j 列の行列を作成する．行列内の要素 (i, j) には a_i と b_j が対応している．次に，行列の要素 (i, j) へ，a_i と b_j の局所距離 $d(i, j)$ を格納する．これを行列の全要素について行う．なお，局所距離にはユークリッド距離を用いる．すなわち，$d(i, j)$ は (1) 式で表される．

$$d(i, j) = ||a_i, b_j|| \tag{1}$$

最後に，行列内の局所距離を用いて，各要素での累積距離 $D(i, j)$ を求める．$D(i, j)$ は (2) 式で表される．

$$D(i, j) = \min \begin{cases} D(i-1, j) + d(i, j) \\ D(i, j-1) + d(i, j) \\ D(i-1, j-1) + 2d(i, j) \end{cases} \tag{2}$$

これを最後まで計算することで，パターン間の最小累積距離 $D(I, J)$ を求めることが出来る．最小累積距離は，2 つのパターンが類似しているほど小さい値となる．

また，$D(I, J)$ を求める過程で，累積距離が最小になる際に，時系列パターンのどことどこの要素が類似していたかを表す最短経路が求まる．DP マッチングの最短経路の例を**図 1** に示す．

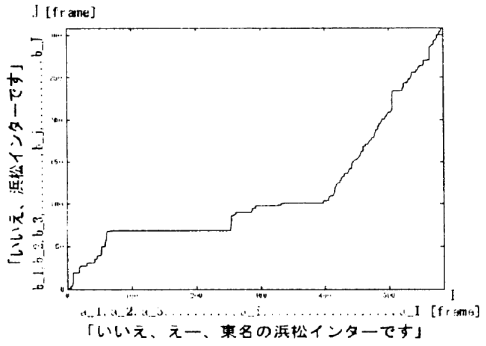

図 1 DP マッチングの最短経路の例
([6] pp.3 fig.3 より転載)
Fig. 1 The example of DP Matching
(Reprinted from [6] pp.3 fig.3)

図 1 のように I と J を各軸にとると，経路は縦，横，斜めの 3 方向に進むことがわかる．経路が斜めに進む区間は i と j の対応が変化しない，すなわちそこの区間ではパターンが類似していることを表している．図 1 では「いいえ」と「浜松インターです」の区間で経路が斜めに進んでいる

ことがわかる．この経路が斜めに進む区間のことを線形区間と呼ぶ．

今回の評価では各発話を 10 ミリ秒間隔で区切り，各区間で音声特徴量を抽出して時系列パターンにした．音声特徴量は，12 次元の MFCC 特徴量にエネルギーを加えた計 13 次元の特徴量ベクトルとした．判定したい 2 つの発話の音声特徴量に対して DP マッチングを行い，最小累積距離およびその際の最短経路を求め，以下のような手法で判定を行った．

3.1.1 DP マッチングの最小累積距離 nomi による判定

いくつかの論文では繰り返し発話の検出のために，DP マッチングの最小累積距離を用いる手法が採用されている [6][9][10]．その中で，我々は今井らの手法 [9] を比較に用いる．DP マッチングの結果の最小累積距離は，2 つのパターンが類似しているほど小さい値となることを利用して，最小累積距離が閾値以下であれば繰り返し発話である，閾値よりも大きければ繰り返し発話でないと判定する．閾値は，学習データを判定した際に，F 値が最も大きくなるように値を設定する．

3.1.2 DP マッチングの線形区間での距離による判定

実際の発話では，図 1 のように繰り返し部分とは無関係な部分が多く存在すると想定される．矢野らはそれを考慮して線形区間を取り出して判定を行っている [6]．我々もそれを比較手法の 1 つとして試みる．線形区間の長さが最小区間長未満である場合，繰り返し発話でないと判定する．線形区間の長さが最小区間長以上であり，線形区間での累積距離が閾値以下であれば繰り返し発話である，閾値よりも大きければ繰り返し発話でないと判定する．閾値は，線形区間の長さが最小区間長未満であるものを除いた学習データを判定した際に，F 値が最も大きくなるように値を設定する．また，最小区間長は [6] 内で使用されている 25 フレーム (250ms) とした．

3.1.3 DP マッチングの結果から特徴量を抽出することによる判定

川井らは，DP マッチングの結果から特徴量を抽出して判定を行っている [8]．

本稿では以下の 4 種の特徴量に加え，

- 最小累積距離
- 線形区間の総数
- 横方向の最大連続移動数
- 縦方向の最大連続移動数

以下の 2 種も併せて使用する．

- 片方の発話のフレーム数
- もう片方の発話のフレーム数

これら計 6 種類の特徴量を入力としたランダムフォレストで繰り返し発話であるかどうかを分類させる．重なり度および編集距離は使用しない．

3.2 音声認識結果の文字列による判定

今回の実験では，Microsoft 社の Bing Speech API と Google 社の google cloud speech API を用い，評価データについて音声認識を行った．

どちらの API でも，音声認識が出来た場合は認識結果の文字列が，不可能であった場合にはそれを示す値が返って来る．矢野らの手法では認識結果の複数の候補を用いて判定を行っている [6] が，今回使用した API 用のライブラリでは任意の数の認識候補を得ることができない．そのため，繰り返し発話かどうかを判定する 2 つの発話について，音声認識の結果としてどちらかでも音声認識不可能と返された場合，それは発話でない，すなわち繰り返し発話でないと判定した．また，両方とも認識できた場合には，結果の文字列が完全一致した場合は繰り返し発話であると判定し，そうでない場合は繰り返し発話でないと判定した．

4. 提案手法

ここでは，提案手法である LSTM を用いた 2 つの提案手法について説明する．

LSTM(Long short-term memory) は，1997 年に Hochreiter らによって提案された RNN （回帰型ニューラルネットワーク）の 1 種である [11]．RNN は前に入力した情報を記憶することが可能であり，特に時系列データに対して高い性能を発揮する．LSTM は従来の RNN で発生していた勾配消失問題に対処するために発案され，RNN よりもより長期の入力情報を記憶しておくことが可能である．

LSTM の応用例として，例えば Wen らは自然言語生成に LSTM を導入することにより，従来手法よりも有益で自然らしい，客観的な評価でも高い性能を持つ言語応答を生成可能とした [12]．また，Li らはサーバの消費電力の時系列データ（波形）からそのサーバ内で動作しているプログラムの推測をする際に，LSTM を使用している [13]．

4.1 提案手法 A

提案手法 A は，LSTM に MFCC 特徴量を入力する手法である．

LSTM は事前に入力した情報を記憶することが可能である．始めに入れた発話の特徴量を入力し LSTM に記録させておくことで，後からもう 1 つの発話の特徴量を入れた際に，事前の発話の情報と比較することで繰り返し発話かどうかを判定することが期待できる．ただし，後から入力する発話の方が先に入力する発話より有益な情報を保持している可能性がある．そのため，入力する発話の順番を入れ替えた状態と計 2 種類で学習および判定を行う．

提案手法 A のブロック図を**図 2** に示す．

4.1.1 音声特徴量の作成

まず，各発話を 10 ミリ秒間隔で区切り，各区間で音声特徴量を抽出して時系列パターンにする．音声特徴量は，12

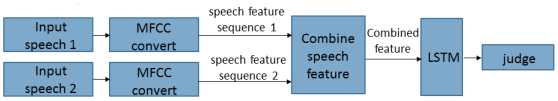

図 2 提案手法 A のブロック図
Fig. 2 The Block Diagram of Proposed Method A

次元の MFCC 特徴量にエネルギーを加えた計 13 次元の特徴量ベクトルとした．

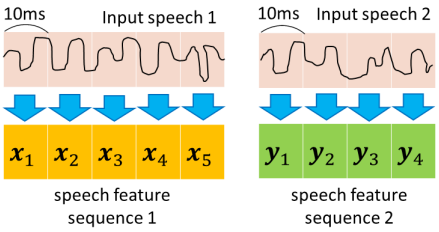

図 3 音声特徴量の作成
Fig. 3 Making Speech Feature

4.1.2 音声特徴量の結合

次に，**図 4** のように 2 つの音声特徴量を結合して 1 つにする．

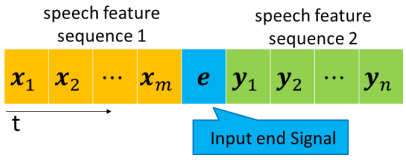

図 4 音声特徴量の結合
Fig. 4 Combining Speech Feature

具体的には，片方の発話の MFCC 特徴量 x_t の後に，入力終了を表す信号 e を，さらにその後にもう片方の発話の MFCC 特徴量 y_t を結合するようにした．これを，発話 1 の特徴量を前に，発話 2 の特徴量を後ろにつけたものと，発話 2 の特徴量を前に，発話 1 の特徴量を後ろにつけたもの，計 2 つの特徴量を作成する．

4.1.3 LSTM ネットワークへの入力

最後にこの結合した特長量をフレームごとに次のような構成の LSTM ネットワークに入力し，その際の出力から繰り返し発話かどうかの判定を行う．

- 入力層：13 次元の音声特徴量を入力するための，線形ニューロン 13 個
- 中間層：LSTM2 層，1 層あたり LSTM ニューロン 500 個
- 出力層：活性化関数を sigmoid 関数とした，判定用のニューロン 1 個

この LSTM ネットワークに対し，**図 5** のように結合した特徴量をフレーム毎に入力していく．

入力終了信号を入力する際には，入力層の全ニューロンに-1 を入力する．後に入力する方の音声の特徴量部分 y_t を入力した際の LSTM ネットワークからの出力 z_t により，入力した 2 つの発話が繰り返し発話かどうかの判定を行う．これを，もう 1 つの結合した特徴量でも行い，計 2 パターンの y_t と z_t の組を作成して最終的な判定を行う．

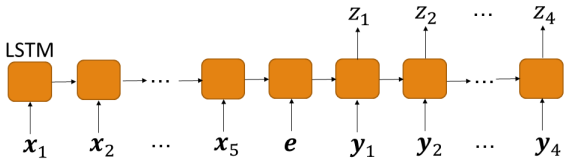

図 5 LSTM ネットワークの入出力
Fig. 5 The Input and Output of LSTM Network

4.1.4 学習方法

学習時には y_t を入力する際，繰り返し発話であり，かつ発話部分のフレームを入力したとき対応する出力 $z_t = 1$，繰り返し発話でないまたは環境音部分のとき $z_t = 0$ となるように学習させる．

学習を行うにあたり，ドロップ率 50%，バッチ数 300 のミニバッチ学習とした．また，誤差の算出には cross entropy を用い，Adam 法で最適化を行った．その際，勾配に対して閾値 5 でクリッピングを行った．

なお，学習時には，発話音声内の発話部分とそうでない環境音部分を判別する必要がある．発話部分の音声は環境音よりも大きいと想定し，音量が一定値以上である部分を発話部分，一定値未満である部分を環境音部分として学習を行った．

4.1.5 判定方法

2 つの結合した特徴量のうち 1 つを LSTM ネットワークに入力して z_t が 1 を出力した回数を記録する．y_t のフレーム数との比が閾値以上であれば，その発話のペアは繰り返し発話であると判定する．一方，y_t のフレーム数との比が閾値未満である場合は，もう 1 つの結合した特徴量を LSTM ネットワークに入力して同様の判定を行う．y_t のフレーム数との比が閾値以上であればその発話のペアは繰り返し発話であると判定し，y_t のフレーム数との比が閾値未満であればその発話のペアは繰り返し発話でないと判定する．

閾値について，どの程度発話が一致していれば繰り返し発話とみなすかは，要求されるタスクや環境によって異なると考えられる．今回は閾値を 0.5，すなわち発話の半分以上が類似していれば繰り返し発話であると判定した．

4.2 提案手法 B

提案手法 B は，LSTM に DP マッチングの結果を入力する手法である．

前述のとおり，川井らは DP マッチングの結果から特徴量を抽出し判定を行っている [8]．したがって，ディープ

ニューラルネットワークを用いて従来手法よりも適した特徴量を抽出して分類すれば，より高い性能を発揮することが期待される．

提案手法 B のブロック図を**図 6** に示す．

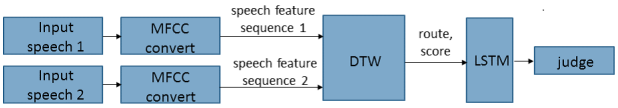

図 6 提案手法 B のブロック図
Fig. 6 The Block Diagram of Proposed Method B

4.2.1 音声特徴量の作成

まず，提案手法 A と同様の方法で 13 次元の音声特徴量を作成する．

4.2.2 音声特徴量についての DP マッチング

次に，比較する 2 つの音声の特徴量について DP マッチングを行い，全体の累積距離が最小になるときの経路について，進んだ方向および進んだ地点までの累積距離を記録しておく．方向については 3 次元のベクトルを用意し，各要素に縦方向，斜め方向，横方向と対応付けして，方向と対応する要素を 1 に，それ以外の要素は 0 と記録する．例えばベクトルの要素と経路の対応を (縦方向, 斜め方向, 横方向) とした際，縦方向に進んだ場合は (1,0,0) と記録する．

4.2.3 LSTM ネットワークへの入力

最後にこの特長量をフレームごとに LSTM ネットワークに入力し，その際の出力から繰り返し発話かどうかの判定を行った．

我々はこの LSTM を用いて，次のような構成の LSTM ネットワークを作成した．

- 入力層：4 次元の特徴量 (累積距離 1 次元と経路 3 次元) を入力するための，線形ニューロン 4 個
- 中間層：LSTM2 層，1 層あたり LSTM ニューロン 50 個 [*1]
- 出力層：恒等関数を活性化関数とした，判定用のニューロン 1 個

この LSTM ネットワークに対し，特徴量をフレーム毎に入力していく．

4.2.4 学習方法

学習時には y_t を入力する際，繰り返し発話であり，かつ経路が斜め方向であるフレームを入力したとき対応する出力 $z_t = 1$，繰り返し発話でないまたは経路が斜め方向でないとき $z_t = 0$ となるように学習させる．

学習を行うにあたり，ドロップ率 50%，バッチ数 300 のミニバッチ学習とした．また，誤差の算出には mean squad error を用い，Adam 法で最適化を行った．その際，勾配に対して閾値 5 でクリッピングを行った．

[*1] 予備的に 500 個でも試したが，大きな変化は見られなかったため，50 とした．

4.2.5 判定方法

2 つの結合した特徴量のうち 1 つを LSTM ネットワークに入力して z_t が 1 を出力した回数を記録する．実際に斜め方向であった数との比が閾値以上であれば，その発話のペアは繰り返し発話であると判定した．一方，実際に斜め方向であった数との比が閾値未満であればその発話のペアは繰り返し発話でないと判定した．

閾値について，提案手法 A と同様に 0.5 とした．

なお，t=0 の際の進んだ方向は定義されていない．そのため，その際の特徴量入力時は斜め方向に進んだものとし，t=0 のフレームは無視した．

5. 評価実験

5.1 評価尺度

正例に対して負例の割合が大きいことから，評価時の指標として F 値を使用した．

F 値は，正と予測したもののうち実際に正であった割合である適合率 (presicion) と，正であるもののうち正と予測できた割合である再現率 (recall) の，調和平均をとったものである．

F 値を求めるには，まず評価データについているラベル (label) および手法での判定結果 (prediction) に基づき，**表 2** の TP，FP，FN，TN それぞれに該当するインスタンス数を求める．

表 2 評価結果の分類
Table 2 Classification of Evaluation Result

		予測結果	
		正例	負例
真の結果	正例	TP	FP
	負例	FN	TN

その後，TP, FP, FN を以下の式に代入することで F 値を求めることが出来る．

$$precision = \frac{TP}{TP + FP} \tag{3}$$

$$recall = \frac{TP}{TP + FN} \tag{4}$$

$$Fmeasure = \frac{2recall \cdot precision}{recall + precision} \tag{5}$$

F 値は正例 (繰り返し発話である) についてと負例 (繰り返し発話でない) についての 2 種類について求めることが可能であるが，今回は正例についての F 値のみ求めた

なお，手法の妥当性を検証するために 10 分割交差検証を行った．全インスタンスを 10 分割し，その中の 1 つを評価データ，残りの 9 つを学習データとした．分割した各データがそれぞれ評価データとなるよう，計 10 種類の学習データと評価データのセットを作成した．また，正例と負例のインスタンス数の比が 1:1 になるように調整した学

習データを使用した場合と，インスタンス数を調整していない学習データを使用した場合の 2 パターンについて評価を行った．

ただし，時間の都合上各提案手法に関して，学習データを調整した・していないで完全に対照な評価が現段階でできていない．提案手法 A について，学習データを調整していない場合のモデルは前述のとおりであるが，学習データを調整した場合のモデルは以下の点が異なる．

- LSTM ネットワークの出力層について，活性化関数が恒等関数
- 損失関数が mean squad error

また，未調整の学習データを使用した場合の提案手法 A について，10 分割交差検証を行えていない．提案手法 B については，インスタンス数を調整していない学習データを使用したパターンの評価は行えていない．

5.2 提案手法の epoch 数について

epoch 数，すなわち学習データを何周分学習するかは，10 分割したうちの始めの 1 セットについて，epoch 数を 1 から 50 の 50 パターンで評価を行い，その中で最も良い F 値を出した epoch 数を用いる事とした．なお，未調整の学習データを使用した場合の提案手法 A について，時間の都合上 epoch 数を 1 から 9 までででしか評価を行っていない．

6. 評価結果

6.1 提案手法 A の epoch 数の決定

提案手法 A について，epoch 数を決定するために，10 分割交差検証の 1 回目を epoch 数を 1 から 9 に変化させて評価を行った．評価を行った結果について，調整した学習データを使用した場合を**図 7** に，未調整の学習データを使用した場合を**図 8** に示す．

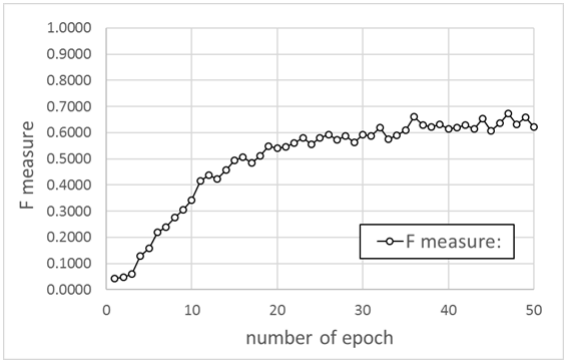

図 7 学習データを調整した場合の epoch 数と F 値の関係
Fig. 7 The Relation between Number of Epoch and F Measure with Adjusting Training Data

この結果，学習データを調整している場合，epoch 数が 47 のとき F 値が最大値 0.672 となった．一方，学習データを調整していない場合，epoch 数が 9 のとき F 値が最大値 0.975 となった．そのため，提案手法 A について 10 分割交差検証をする際は，データ数を調整している場合 epoch 数を 47 として評価を行った．

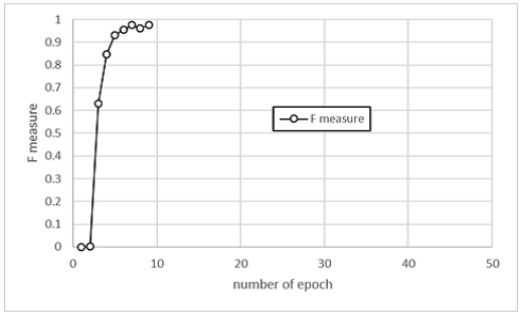

図 8 学習データを調整していない場合の epoch 数と F 値の関係
Fig. 8 The Relation between Number of Epoch and F Measure with No Adjusting Training Data

6.2 提案手法 B の epoch 数の決定

提案手法 B についても，epoch 数を決定するために，10 分割交差検証の 1 回目を epoch 数を 1 から 50 に変化させて評価を行った．評価を行った結果について，調整した学習データを使用した場合を**図 9** に示す．

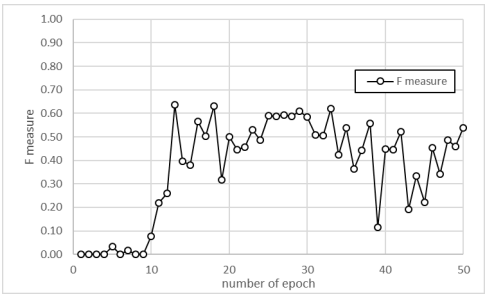

図 9 学習データを調整した場合の epoch 数と F 値の関係
Fig. 9 The Relation between Number of Epoch and F Measure with Adjusting Training Data

この結果，epoch 数が 13 のとき F 値が最大値 0.637 となった．そのため，提案手法 B について 10 分割交差検証をする際は，epoch 数を 13 として評価を行った．

6.3 各手法の評価結果

各手法の評価結果のまとめについて，調整した学習データを使用した場合を**表 3** に，未調整の学習データを使用した場合を**表 4** に，示す．ただし，majority baseline は全てのインスタンスを負例として評価した場合の 10 分割交差検証の結果を表している．また，学習データを調整していない場合の提案手法 A の結果については，F 値が安定したと思われる epoch 数が 5 から 9 のときの各値の平均を載せる．

表 3 学習データを調整した場合の各手法の評価結果の比較
Table 3 Comparison with each Evaluation Result with Adjusting Training Data

手法	正解率	適合率 (正例)	再現率 (正例)	F 値 (正例)
majority baseline	0.978	-	-	-
DP マッチング (最小累積距離)	0.742	0.068	0.857	0.127
DP マッチング (線形区間)	0.907	0.153	0.713	0.251
DP マッチング (特徴量抽出)	0.941	0.260	0.922	0.406
音声認識結果の文字列 (bing)	0.983	0.988	0.222	0.363
音声認識結果の文字列 (google)	0.981	0.998	0.146	0.255
提案手法 A	0.974	0.458	1.000	0.628
提案手法 B	0.982	0.756	0.286	0.415

表 4 学習データを調整していない場合の評価結果の比較
Table 4 Comparison with each Evaluation Result with No Adjusting Training Data

手法	正解率	適合率 (正例)	再現率 (正例)	F 値 (正例)
majority baseline	0.978	-	-	-
DP マッチング (最小累積距離)	0.966	0.277	0.353	0.311
DP マッチング (線形区間)	0.972	0.326	0.276	0.299
DP マッチング (特徴量抽出)	0.991	0.861	0.695	0.769
音声認識結果の文字列 (bing)	0.983	0.988	0.222	0.363
音声認識結果の文字列 (google)	0.981	0.998	0.146	0.255
提案手法 A	0.998	0.950	0.968	0.959

7. まとめと今後の課題

7.1 まとめ

現在，音声対話エージェントに対して注目が高まっている．しかし，今日の音声認識は精度が高くなっているものの，例えば音声認識部分でノイズや環境の変化を受ける，システムの辞書に登録されていない単語，すなわち未知語を検知および理解することが出来ない等，発話内容の誤認識は避けられないのが現状である．

一方，ユーザは発話した内容が誤認識されたと考えた際，その発話を繰り返し行う傾向がある．つまり，この繰り返し発話をシステムが検出することが出来れば，システムが誤認識してしまったことを検出する手がかりの一つとなる．そこで，本研究は繰り返し発話を検出することで，システムの誤認識検出に貢献することを目的とする．

提案手法では，判定の基準として，音声認識結果の文字列ではなく，音声波形の類似性を用いた．また，近年ディープニューラルネットワークを用いた機械学習が分野を問わずに活用され，成果を出していることから，今回の目的にもディープニューラルネットワークが使用できると考え，時系列データを扱うことが出来る LSTM を使用することにした．

評価対象として，合成音声に単語を発話させたものを 1 つの単語に付き 25 個，44 単語分，計 1100 個を作成した．この評価対象に対して，提案手法である 2 つの発話の音声特徴量を結合して LSTM に入力する手法と，DP マッチングの結果を LSTM に入力する手法，従来手法である DP マッチング，音声認識結果の文字列を比較する手法で 10 分割交差検証を行い各手法の性能の比較を行った．その結果，特に 2 つの発話の音声特徴量を結合して LSTM に入力する提案手法の方が従来手法よりも繰り返し発話検出の性能が高くなることを確認することができた．

7.2 今後の課題

まず提案手法については，判定方法に特にアドホックな部分がある．他の判定方法も検討したい．

次に，提案手法について学習データを調整する・しないで結果を比較できるように評価結果をそろえることが必要である．図 9 のように学習中に F 値が不規則に変化することがあったが，この原因も結果を比較することで考察をすることが可能になると考えられる．

また，従来手法である DP マッチングの線形区間での距離による判定における最小区間長，提案手法 A での繰り返し発話かどうか判定するフレーム数の比の閾値，提案手法の各種パラメータ等，調整を行っていない．そのため，これらの値を調整することで性能が向上する可能性がある．川井ら [8] の手法も，参考にしたが，彼らが提案している手法そのものを実装して比較出来ていない．

特徴量については，MFCC 特徴量の代わりにログフィルタバンク特徴量の使用が挙げられる．より生に近い特徴量であるログフィルタバンク特徴量を使用することで，より最適な特徴量を LSTM が自動的に抽出，判定することにより，より性能が高い手法になる可能性がある．

評価データについては，人工的に編集を行った合成音声ではなく実際の発話音声を用いた評価の追加が必要である．

Levitan らの手法では，異なる手法を組み合わせた評価方法を提案している [7] ことから，今回の提案手法も他の手法と組み合わせることで性能の向上が期待出来る．

謝辞 本研究に際して、データを共有してくださった（株）Nextremer の川井 雄太氏と谷川 晃大氏に深く御礼申し上げる．

参考文献

[1] Bellegarda, J. R.: Spoken language understanding for natural interaction the Siri experience, *In Proc. IWSDS*, pp. 3–14 (2012).
[2] 東中竜一郎，貞光九月，内田　渉：しゃべってコンシェルにおける質問応答技術，NTT 技術ジャーナル，Vol. 25, No. 2, pp. 56–59 (2013).
[3] 久保陽太郎：音声認識のための深層学習，人工知能，Vol. 29, No. 1, pp. 626–634 (2014).
[4] 柏木陽佑，齋藤大輔，峯松信明，広瀬啓吉：雑音環境下音声認識のためのディープニューラルネットワークを用いた識別的区分線形変換，電子情報通信学会論文誌 D，Vol. J99-D, No. 3, pp. 255–263 (2016).
[5] Cevic, M., Weng, F. and Lee, C.-H.: Detection of Repetitions in Spontaneous Speech in Dialogue Sessions, *Proc. INTERSPEECH*, pp. 471–474 (2008).
[6] 矢野浩利，北岡教英，中川聖一：対話システムにおける言い直し・否定表現に着目した訂正発話の検出，情報処理学会研究報告，Vol. 2005-SLP-55, pp. 95–100 (2005).
[7] Levitan, R. and Elson, D.: Detecting Retries of Voice Search Queries, *Proc. ACL*, pp. 230–235 (2014).
[8] 川井雄太，藤田寛泰，谷川晃大，山下　峻，船越孝太郎：応答義務推定の補助としての繰り返し発話検出，情報処理学会研究報告（第 116 回音声言語情報処理研究会）(2017).
[9] 今井裕志，井ノ上直己，橋本和夫，米山正秀：未知語処理のための繰り返し音声検出手法，技術報告，社会法人電子情報通信学会 (1999).
[10] Kitaoka, N., Kakutani, N. and Nakagawa, S.: Detection and Recognition of Correction Utterance in Spontaneously Spoken Dialog, *Proc. EUROSPEECH 2003*, pp. 625–628 (2003).
[11] Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, *Neural Computation*, Vol. 9, pp. 1735–1780 (1997).
[12] Wen, T.-H., Gasic, M., Mrksic, N., Su, P.-H., Vandyke, D. and Young, S.: Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems, *Proc. EMNLP*, pp. 1711–1721 (2015).
[13] Li, Y., Hu, H., Wen, Y. and Zhang, J.: Power Data Classification: A Hybrid of a Novel Local Time Warping and LSTM, arXiv.org (online), available from ⟨https://arxiv.org/pdf/1608.04171.pdf⟩ (accessed 2017-01-12).