

音声による地名入力タスクにおける誤認識時の言い直し発話の検出と認識

角谷 直子[†] 北岡 教英[†] 中川 聖一[†]

[†] 豊橋技術科学大学 情報工学系 〒441-8580

E-mail: †{naoko,kitaoka,nakagawa}@slp.ics.tut.ac.jp

あらまし 近年、音声認識をベースとしたインターフェースを備えたカーナビゲーションシステムの実用化が進んでいる。しかし、音声認識を用いる場合、誤認識は避けられない。ここで、誤認識の訂正のためにユーザが誤認識された部分を言い直した場合に、それが言い直しの音声であるとシステムが判定できれば、誤認識からの回復が容易になると考えられる。本稿では、DP マッチングによるワードスポッティングと認識候補の重なり度を用いて、カーナビ地名入力タスクでよく用いられる部分的な言い直し発話を検出する手法を提案し、言い直し発話に対する検出率 97.0%、それ以外の発話に対する棄却率 93.7%と高い検出精度を得ることができた。更に、言い直し判定結果を用いて認識率を向上させる方法として、判定結果から語彙を限定して認識する方法とリアルタイム認識が可能であるリスコアリングによる方法を提案し、両者において、言い直し検出による認識率向上の効果を確認した。

キーワード カーナビゲーションシステム, 音声認識, 言い直し, 言い直し発話検出, ワードスポッティング

Detection and Recognition of Repaired Speech for Misrecognition in Location Name Input Task

Naoko KAKUTANI[†], Norihide KITAOKA[†], and Seiichi NAKAGAWA[†]

[†] Toyohashi University of Technology Dept. of Information & Computer Sciences

E-mail: †{naoko,kitaoka,nakagawa}@slp.ics.tut.ac.jp

Abstract Recently, car navigation systems with a speech interface have been developed. When we communicate with computers through speech recognition, one can not avoid misrecognition and it is difficult to recover from this condition because the interface is only in the initial state of the art. Detection of user's repetition of a misrecognized part can make it easier. We propose a method to detect partial repetition of misrecognized speech using a word spotting technique based on DTW (dynamic time warping) and N -best hypotheses overlapping measure. We achieved 97.0% detection accuracy for repetitions and 93.7% rejection accuracy for non-repetitions. Next, we tried to improve recognition accuracy using detection. Using the choice of vocabulary setup based on the detection, we achieved improvement in recognition performance in adverse conditions with SNR of 7-13 dB from 29.0% to 35.1% for repaired speech and from 55.5% to 59.4% for non-repetitions. When we employed a rescoring strategy for real time processing, we achieved a 33.5% recognition rate for repaired speech and 57.9% for non-repetitions.

Key words car navigation system, speech recognition, repaired speech, repaired speech detection, word spotting

1. はじめに

近年、音声認識をベースとしたインターフェースを備えたカーナビゲーションシステムの実用化が進んでいる。運転中という特殊な環境で、複雑なシステムを操作するためのインターフェースとして、音声入力を中心とした音声対話インターフェースが注

目され、多くの製品で採用されている。

コンピュータと人間が音声を通じてコミュニケーションをはかる場合、誤認識は避けられない。しかし、現在は音声インターフェース技術が未熟であるために、誤認識からの回復が困難である。一般に、ユーザはシステムの誤認識に対して、同じ内容の言い直しで対処しようとすることが多い。すなわち、

システムがユーザの言い直しを検出できれば、誤認識からの回復が容易になると考えられる。

現在も言い直し(繰り返し)音声に関する研究はいくつかなされている。言い直し発話の韻律的な特徴についていくつか研究がされており、繰り返し発話は元のユーザ発話に比べてピッチ・継続時間長が大きく発話速度の低下が見られる[5][7]、訂正発話では継続時間が長くなるがパワーやピッチについては、大きな差は見られない[8]、認識誤りの訂正と棄却誤りの訂正では認識誤り訂正の方が継続時間がより長くなる[9]、訂正連鎖においてエラーからより遠い訂正は近い訂正よりピッチ・パワーが大きく継続時間が長くゆっくりで先行ポーズが長い[10]などの報告がある。また、F0とパワーを用いて訂正発話の特徴を分析したところ、誤認識された発話と初回訂正発話の変化は有意ではなかったが、変化のタイプによって被験者を分類し、再分析したところ、有意水準1%で変化が有意であったと報告されている[11]。我々の以前の研究[1]においても、言い直し発話は声の大きさ、声の高さによる抑揚が大きくなっていることが分かっている。

また、言い直し発話の検出についても研究されており、井ノ上・今井らは、未知語処理のための孤立単語の繰り返し音声検出の手法として、(1)認識候補の重なり度による識別手法、(2)認識尤度差による識別手法、(3)パワーの時系列ベクトル間の距離による識別手法、の3通りの手法を提案しており、手法1と手法3を組合せることによって、再現率95.8%、適合率95.9%の検出精度を得ている[3][4]。

一方、カーナビの地名入力タスクにおいては、図1のように地名の下位階層(町名、または、市・町名)のみが誤認識してしまうという現象も少なくなく、その際、ユーザは誤認識箇所のみを言い直すことによって対処しようとすることが多い[1]。

本稿では、このような下位階層のみの誤認識に対する下位階層のみの言い直し発話を検出する方法として、(1)DPマッチングによるワードスポットティング法、(2)認識候補の重なり度を用いた判定方法、を提案する。また、このような地名の下位階層のみの言い直し(以下、単に言い直しという)の検出結果を利用して認識率の向上をはかる。本研究で用いた認識辞書は図2に示すような木構造になっている。認識の際に、言い直し判定のない場合はシステムは言い直しと言い直しでない場合の両方を予測して語彙を設定する必要があり、地名のトップノード(言い直しでない)と前発話の認識結果の全中間ノード(言い直しと仮定)を認識語彙として設定する。しかし、実際にはこれら全てを設定することによる認

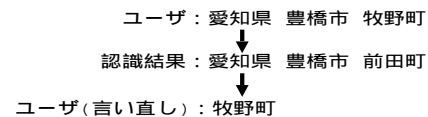


図1 地名入力タスクにおける言い直しの例

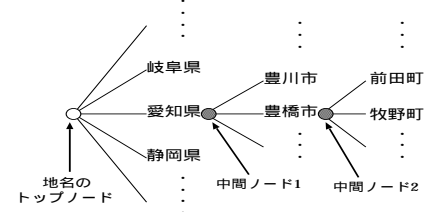


図2 辞書の構造と語彙の設定

識誤りも少なくない(注1)。ここで、言い直しを判定でき、判定結果に従って語彙を限定することができれば、これらの誤認識を回避できる可能性がある。図1の例を考えると、ユーザの言い直し「牧野町」が言い直しであると判定できれば、認識対象として中間ノード2からの語彙に限定することができる。また、言い直しでない判定された発話に対しては地名のトップノードからの語彙に限定することができる。このように、状況に応じて認識対象語彙を設定すれば、認識率の向上につながると考えられる。

2. 言い直し検出方法

2.1 DPマッチングによる判定

言い直した音声(例:「牧野町」)が直前に発声した音声(例:「愛知県豊橋市牧野町」)に含まれているかどうかを調べ、言い直しかどうかを判定する。

方法としては、直前の発話音声Aの*i*フレーム目 a_i と現発話音声Bの*j*フレーム目 b_j とのケプストラム距離 $g(i, j)$ に基づくDPマッチングによって、ワードスポットティングを行う。すなわち、時系列パターンの中から部分パターンを抽出する。まず、図4中の式で定義するような各(*i, j*)に対する a_i と b_j のユークリッド距離(局所距離) $d(i, j)$ を求め、*j*方向にパスが進むごとに局所距離 $d(i, j)$ が加算される。図3のようなDPパスを用いて(*I, J*)までの最小累積距離 $g(I, J)$ を求める。このDPパスは*j*方向に照合が進む際に距離が加算されていく。その結果、DPマッチングのスコアが閾値よりも小さく、かつ、照合開始位置 $B(I, J)$ が最初の発話の認識結果の地名の境界とほぼ一致(±フレーム)する場合には、その部分から言い直したと判定し、それ以外は別の語を発声した、すなわち言い直しでない判定する。また、終端付近の発声が曖昧になったり、

(注1): 後に述べる実験において、常に全てのノードを設定した場合、言い直し発話を言い直し発話でないとして誤認識したものが22.2%、言い直しでない発話を言い直し発話として誤認識したものが2.5%あり、特に前者に問題があることが分かる。

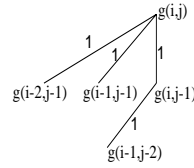
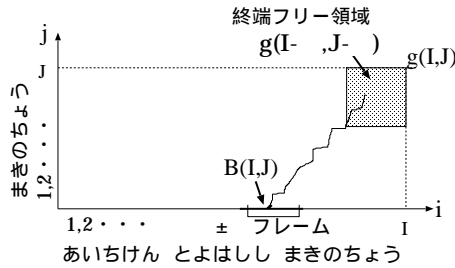


図 3 DP パスと重み



*for $i = 1, 2, \dots, I, j = 1, 2, \dots, J$

$$d(i, j) = \sum_{k=1}^D (a_{ik} - b_{jk})^2 \quad (D: \text{ケプストラム係数の次元数})$$

*for $i = 1, 2, \dots, I$

$$g(i, 1) = d(i, 1) \quad g(i, 0) = 0$$

$$B(i, 1) = i \quad B(i, 0) = i + 1$$

*for $j = 2, 3, \dots, J$

$$g(1, j) = \infty$$

*for $j = 1, 2, 3, \dots, J$

$$g(0, j) = \infty$$

*for $i = 1, 2, \dots, I, j = 1, 2, \dots, J$

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

$g(i-2, j-1) + d(i, j)$ のとき、 $B(i, j) = B(i-2, j-1)$

$g(i-1, j-1) + d(i, j)$ のとき、 $B(i, j) = B(i-1, j-1)$

$g(i-1, j-2) + d(i, j-1) + d(i, j)$ のとき、 $B(i, j) = B(i-1, j-2)$

図 4 ワードスポットティング

表 1 音声分析条件

サンプリング周波数	12 kHz
フレーム	Hamming 窓
フレーム長	21.33 ms
フレーム周期	8 ms

自動検出された音声の終端のずれが大きく、うまくマッチングしない可能性があるため、両音声の終端を固定せず、ある程度フリー（× フレーム）にするなどの改良法を適用する（図 4 参照）[1][2]。

ここでは、音声の特徴パラメータとして、10 次元 LPC メルケプストラムを用いた。表 1 に音声分析条件を示す。

2.2 認識候補の重なり度による判定

直前の発話 A、現発話 B に対して、言い直し、言い直しでない両方を予測した辞書設定で認識を行い、N-best 候補を求める。B の N-best 候補と A の N-best 候補とを比較し、同一の候補が含まれているかどうかを調べる。同一の候補が含まれている数を「重なり候補数」とし、重なり度としては以下のように定義する[3][4]。（中間ノードからの認識結果はトップノードからの認識結果となるように後処理し

ている。）

$$\text{重なり度} = \frac{2 \times \text{重なり候補数}}{N - \text{best 候補数} + \text{直前の発話の } N - \text{best 候補数}}$$

よって、言い直し発話の場合は重なり度が大きく、言い直しでない発話の場合は重なり度が小さくなるということが予想される。そこで、この重なり度を言い直し判定尺度として用いる。また、重なり度に対して、各候補の N-best 中の順位を用いて以下のような重み W を付け、上位の候補の重なりを重視することも試みた。

$$W = \left(\frac{1}{\text{前発話の順位}} \times \frac{1}{\text{現発話の順位}} \right)^{1/k} \quad (1)$$

3. 評価実験

3.1 評価データ

カーナビの音声インターフェースを簡単に実現したプロトタイプシステムを用いて被験者実験を行った。システムの認識語彙は全国の地名及び施設名、約 180,000 語（注：図 1 のような「愛知県豊橋市牧野町」も 1 語とする）である。認識結果は合成音声による提示と同時に画面上に文字でも表示される。

被験者は音声対話システムの使用経験のない 9 名（男性 7 名、女性 2 名）である。約 50 個の地名を正しく認識されるまで入力してもらい、地名の下位階層のみが誤認識した場合は、誤認識した部分のみを言い直せることを教示した。

また、実際の悪環境を模擬するためと言い直し発声を多く収集するために、走行自動車内のノイズをスピーカから発生させながら実験を行った。S/N 比は話者によって異なるが、おおよそ 7~13 dB 程度である。各地名入力における初回の発話の認識率は 58.4%であった。なお、S/N 比が 40 dB 程度以上ある場合は、認識率は 90%を超える。

以下の評価実験では、これにより得られた下位階層のみの言い直し発話 131 及び言い直しでない発話 767 に対して言い直し判定の評価を行う。

3.2 評価尺度

評価尺度として、言い直し発話を正しく言い直しと判定した割合（検出率）と言い直しでない発話を正しく言い直しでないと判定した割合（棄却率）を以下のように定義する。

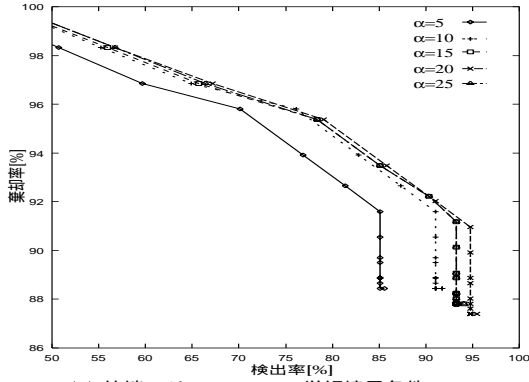
$$\text{検出率} = \frac{\text{正しく言い直しと判定された数}}{\text{言い直し発話の数}}$$

$$\text{棄却率} = \frac{\text{正しく言い直しでないと判定された数}}{\text{言い直しでない発話の数}}$$

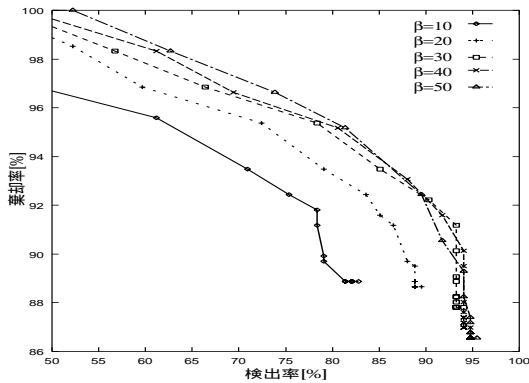
3.3 個々の手法による判定結果

3.3.1 DP マッチングによる判定結果

得られた DP スコアに閾値を設け、閾値を変化させたときの結果を図 5 に示す。(a) は終端フリー領域 30 × 30(240ms) フレームで固定し、単語境界条



(a) 終端フリー 30 × 30, 単語境界条件 ±



(b) 単語境界条件 ± 20, 終端フリー ×

図 5 DP マッチングによる判定性能

件を $\alpha = 5, 10, 15, 20, 25$ と変化させたときの結果、(b) は単語境界条件 ± 20 フレームで固定し、終端フリー領域を $\beta = 10, 20, 30, 40, 50$ と変化させたときの結果である。横軸は検出率、縦軸は棄却率で、右上に近づく程、判定性能が高いことを示している。終端をフリーにする条件と単語境界条件を導入することによって、判定性能が向上した。単語境界条件は ± 20 フレーム、終端フリー領域は 30×30 フレームとした場合が最も良く、検出率と棄却率の平均が最も高くなるように閾値を設定すると、検出率 94.6%、棄却率 91.6% が得られた。

3.3.2 認識候補の重なり度による判定結果

ここでは 200-best を用いた。ただし、アルゴリズム上、必ずしも 200 候補が得られるわけではなく、平均 67 候補となった。得られた重なり度に閾値を設け、閾値を変化させたときの結果を図 6 に示す。(1) 式の重み W は $k = 4, 8, 16, 32, \infty$ (重みなし) とした。結果は、検出率と棄却率の平均が最も高くなるように閾値を設定すると、 $k = 8$ のとき、検出率 77.6%、棄却率 99.3% であった。棄却率は非常に高いのに対して、検出率は比較的低い結果となった。前者は、言い直してない発話は直前の発話と認識候補がほとんど異なっているという予想に合う結果である。それに対して、言い直しの発話は直前の発話

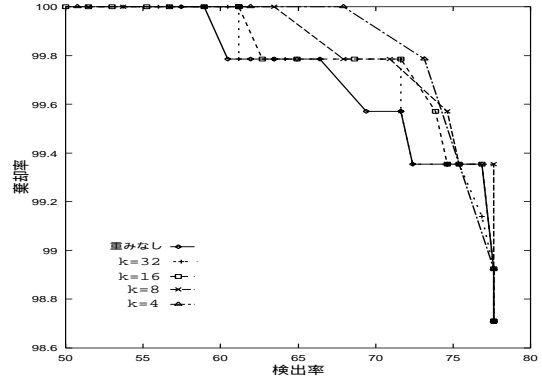


図 6 認識候補の重なり度による判定性能

と認識候補の重なりが大きく異なる場合がしばしばみられることを示している。

3.4 組み合わせによる判定結果

3.4.1 言い直し確率の関数表現

まず、言い直しである確率を定義する。前発話と現発話の組 W に対し、この W が言い直しである場合を $C(W) = 1$ 、言い直してない場合を $C(W) = 0$ 、 W に対するある特徴量 x の値を x_W とする。このとき、 W の特徴量 x の値が x_W であった場合に W が言い直しである確率、すなわち

$$P(C(W) = 1 | x = x_W) \quad (2)$$

を考える。ここで、言い直しである確率は x の関数 $f(x)$ に従っていると仮定する。 $f(x)$ をシグモイド関数

$$f(x) = \frac{1}{1 + \exp(-ax + b)} \quad (3)$$

と仮定し、 $f(x)$ とサンプルの言い直し/言い直してない (1/0) の 2 乗誤差を最小化するようにパラメータ a, b を推定することにより、言い直しである確率の関数を推定する [6]。

3.4.2 単独の特徴量の言い直し確率の関数を用いた近似

判定結果が言い直しである場合、単独の特徴量それぞれから得られる言い直しである確率は両方高くなると考えられる。そこで、2 つの特徴量単独の言い直しである確率の関数 $f_1(x_1), f_2(x_2)$ の積で近似することを考え、ここでは重み付き相乗平均

$$f(x_1, x_2) = f_1(x_1)^w \cdot f_2(x_2)^{1-w} \quad (4)$$

によって 2 つの特徴量に基づく言い直しである確率を表現することを試みる。

3.4.3 言い直し確率の関数を直接推定する方法

言い直しである確率の関数として、

$$f(x_1, x_2) = \frac{1}{1 + \exp(g(x_1, x_2))} \quad (5)$$

を仮定する。 $g(x_1, x_2)$ としては、

- 線形結合 $g(x_1, x_2) = a_1 x_1 + a_2 x_2 + a_3$
- 双 1 次結合 $g(x_1, x_2) = a_1 x_1 x_2 + a_2 x_1 + a_3 x_2 + a_4$

を用いた。これらの係数 a_k を 2 乗誤差最小化により推定した。

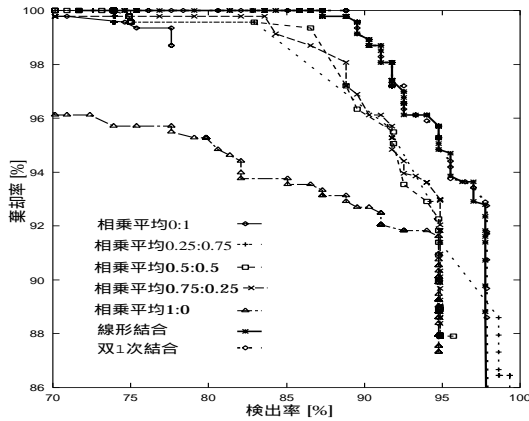


図7 組み合わせによる判定性能

3.4.4 結果

上記の方法を用いて DP スコアと重なり度を組み合わせた場合の判定性能を図7に示す。2つの手法を組み合わせることによって、識別性能が向上することが確かめられた。言い直しである確率の関数を直接推定する方法において、最も高い性能(検出率97.0%、棄却率93.7%)が得られた。また、平均2乗誤差についても、直接推定する方法が最も小さかった。

4. 言い直し判定による効果

認識辞書は、図2に示すような木構造になっている。例えば、直前の発話が「愛知県豊橋市牧野町」で、その結果、牧野町が誤認識され、現発話が「牧野町」の場合を考えると、言い直し判定のない場合は、図2のようにシステムは言い直しと言い直しでない場合の両方を予測して地名のトップノードからと中間ノード1、2から語彙を設定する必要がある。しかし、「牧野町」が言い直しであると判定できれば、地名の中間ノードから(この例の場合中間ノード2から)の語彙に絞ることができる。また、言い直しでない発話に対しても、言い直しでないと言定できれば、地名のトップノードからの語彙に絞ることができる。よって、認識率の向上につながると考えられる。

ここでは、(1)言い直し判定しない場合、(2)100%判定が成功したと仮定した場合、(3)実際の判定性能(検出率97.0%・棄却率93.7%)の場合、の3通りで認識実験を行った。また、実際の判定性能の場合において、言い直し/言い直しでないの2値ではなく、曖昧な部分はグレーゾーンとして、言い直しと言い直しでない場合の両方を予測して語彙を設定する、という実験も行った。グレーゾーンを導入することによって、完全に語彙を絞ることはできなくなるが、判定失敗による悪影響が軽減できると考えられる。

図8に示すように言い直し確率に閾値 γ_1 、 γ_2 を設

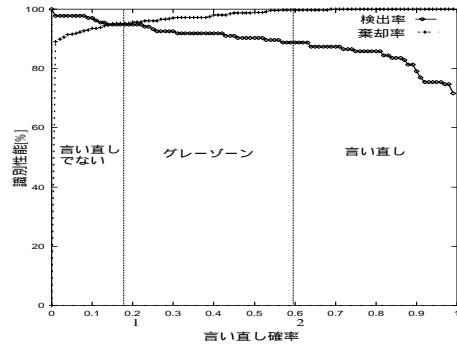


図8 グレーゾーンのある言い直し判定

定し、 γ_1 から γ_2 をグレーゾーンとする。ここでは、(a) $\gamma_1=0.11$ 、 $\gamma_2=0.57$ と (b) $\gamma_1=0.11$ 、 $\gamma_2=0.31$ の2通りの設定で実験を行った。

4.1 判定後に語彙を限定して再認識する方法

発声終了後に判定を行い、その結果をうけて認識を行う方法である。この方法は発声終了後にしか認識が開始できないため、リアルタイム処理ができない。特に、認識候補の重なり度を用いて言い直し判定する場合、1度認識した結果(認識候補)の重なり度から判定し、その判定結果から辞書を絞り、再び認識しなくてはならず、2回の認識処理を要する。

4.2 リスコアリングする方法

言い直しと言い直しでないの両方を予測した認識処理(判定しない場合と同一条件)をリアルタイムに行い、その後判定を行う。認識候補の重なり度を用いる場合にはこの認識結果を用いる。判定結果に従って、認識結果(N-best候補)をリスコアリングする。リスコアリングの方法としては、言い直しと判定されたときは、N-best候補から地名のトップノードからと言い直した部分でない中間ノードから認識されたものは除外し、言い直した部分の中間ノードから認識されたもののみを認識結果とする。このとき、N-best候補中に言い直した部分の中間ノードから認識されたものがない場合は誤認識してしまう。また、言い直しでないと言定されたときは、全中間ノードから認識されたものは除外し、地名のトップノードから認識されたもののみを認識結果とする。この場合も同様に、N-best候補中に地名のトップノードから認識されたものがない場合は誤認識してしまう。この方法では、認識処理を音声入力終了後に行う必要がない。判定処理は計算量が少ないため、ほぼリアルタイムで動作可能である。

4.3 実験結果

それぞれの方法による認識率を表2(言い直し発話について)、表3(言い直しでない発話について)に示す。参考として、100%判定が成功したと仮定した場合の認識率も示す。下位階層のみの言い直しを

表 2 言い直し発話についての認識率

方法	再認識	リスコアリング
言い直し判定なし	29.0%	29.0%
100%判定が成功	37.3%	35.8%
グレーゾーンなし	35.1%	33.5%
グレーゾーン (a)	34.3%	32.8%
グレーゾーン (b)	35.1%	33.5%

表 3 言い直しでない発話についての認識率

方法	再認識	リスコアリング
言い直し判定なし	55.5%	55.5%
100%判定が成功	60.1%	58.9%
グレーゾーンなし	59.3%	57.4%
グレーゾーン (a)	59.9%	58.5%
グレーゾーン (b)	59.4%	57.9%

検出し、語彙を絞ることにより、言い直し音声、言い直しでない音声、両者に対して認識性能向上の効果があることが確認できた。リスコアリングによる認識は、語彙を絞ってから再び認識する方法よりやや劣るものの認識率の向上が得られた。リスコアリングの認識率が再認識の認識率に及ばないのは、リスコアリングする際、言い直しの場合は言い直した部分の中間ノードから、言い直しでない場合はトップノードから認識されたものが N-best 中に現れなかった発話があるためである。(なお、地名のトップノード、各中間ノード別に認識を行っておけば、計算量は3倍になるが、再認識と同一の認識結果が得られる。)

実際の判定性能(グレーゾーンなし)では、100%判定が成功した場合と比べると、向上は小さい。これは、判定に失敗した発話は必ず誤認識されてしまうことが原因である。しかし、グレーゾーンを導入することによって、判定失敗の危険を緩和することができ、認識率も100%判定が成功した場合に近くなった。また、グレーゾーンの設定については、グレーゾーン2よりもグレーゾーン1のほうが認識率が高いことから、なるべく多くの判定に失敗した発話をカバーできるように比較的広めに設定しておいた方がよいことが分かる。なお、言い直し発話の認識率が言い直しでない発話の認識率よりも悪いのは、言い直し発話の方が認識が困難な発話であるからである。

5. ま と め

本稿では、カーナビの地名入力タスクにおける地名の下位階層のみの誤認識に対する下位階層のみの言い直しを検出する手法を提案した。言い直し検出方法として、DP マッチングによるワードスポットティング法と認識候補の重なり度を用いた判定方法について述べ、評価実験を行った。

DP マッチングによる判定では、検出率と棄却率

の平均が最も高くなるように閾値を設定すると、検出率 94.6%、棄却率 91.6%の性能が得られた。認識候補の重なり度による判定では、検出率 77.6%、棄却率 99.3%の性能が得られた。また、言い直し確率を関数で表現することによって、2つの手法を組み合わせさせた結果、関数を直接推定する方法が最も判定性能が向上し、検出率 97.0%、棄却率 93.7%の性能が得られた。

更に、言い直し判定結果を用いて認識率を向上させる方法を検討した。判定結果から語彙を限定して認識する方法と両方の場合を推定して認識した結果をリスコアリングする方法の2つで認識実験を行い、言い直し検出を用いることによって認識率が向上した。リスコアリングによる方法では、語彙を限定して認識する方法よりもやや認識率は劣るものの、リアルタイムで認識が行えるという利点がある。更に、グレーゾーンを導入することによって、100%判定が成功した場合と同程度の効果があることを示した。

今後は、韻律情報も言い直し判定に用いることを考えている。

文 献

- [1] 角谷 直子, 北岡 教英, 中川 聖一:「カーナビの地名入力における誤認識時の訂正発話の分析と検出」, 情処研報, 2001-SLP-37-11(2001.7)
- [2] 角谷 直子, 北岡 教英, 中川 聖一:「カーナビの地名入力における誤認識時の訂正発話の検出と認識」, 電気学会東海支部連合大会, 751, pp.376(2001.11)
- [3] 井ノ上 直己, 今井 裕志, 橋本 和夫, 米山 正秀:「誤認識訂正のための繰り返し音声検出手法」, 電子情報通信学会論文誌, Vol.J84-D-II No.9(2001.9)
- [4] 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」, 電子情報通信学会, SP99-26, pp.1-6(1999.6)
- [5] 平沢 純一, 宮崎 昇, 相川 清明:「質問-応答連鎖からの音声対話システムの誤解の検出」, 電子情報通信学会, SP2000-115, pp.34-41(2000.12)
- [6] 北岡 教英, 赤堀 一郎, 中川 聖一:「認識結果の正解確率に基づく信頼度とリジェクション」, 電子情報通信学会論文誌, Vol.J83-D-II No.11(2000.11)
- [7] 平沢 純一, 宮崎 昇, 相川 清明:「音声対話システムの誤解に対するユーザ応答の分析」, H12 年度春季日本音響学会講演論文集, 3-8-10, pp85-86(2000.3)
- [8] Sharon Oviatt, Margaret MacEachern and Gina-Anne Levow:「Predicting hyperarticulate speech during human-computer error resolution」, Speech-Communication, Vol.24, pp.87-110(1998)
- [9] Gina-Anne Levow:「Adaptation in spoken corrections: Implications for models of conversational speech」, Speech-Communication, Vol.36, pp.147-163(2002)
- [10] Marc Swerts, Diane Litman and Julia Hirschberg:「Correction in Spoken Dialogue System」, IC-SLP2000, Vol.2, pp.615-618(2000)
- [11] 山岸 洋子, 河原 達也:「音声対話システムにおける訂正発話の韻律的特徴の分析」, 人工知能学会研究会, SIG-SLUD-A101-3(2001.6)