社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－ＳＬＰ－69（22）
2007／12／20

生成したテキストのNgramを用いた英語学習者による文法誤りを含んだ発話の音声認識の高精度化

筒井良平† 鈴木基之† 伊藤彰則† 牧野正三†

† 東北大学大学院工学研究科
〒980-8579 宮城県仙台市青葉区荒巻字青葉6-6-5
E-mail: †{tutui,moto,aito,makino}@makino.ecei.tohoku.ac.jp

あらまし 英語学習者がコンピュータを利用して対話練習をできるようなシステムを構築するには，学習者の音声を発話した通りに認識することが必要である．そこで，ここでは，対話時の日本人英語を高精度に認識する方法を検討する．まず，音響モデルに関して検討を行い，よく用いられる3状態HMMよりも4状態HMMや5状態HMMの方が性能がよくなることがわかった．さらに，自動生成したテキストから学習したNgramを言語モデルとして用いて音声認識を行うことで，オートマトンを用いた方法に比べ認識性能が向上した．また，正解文との距離を用いてスコアの再計算を行う手法を提案し，これによって認識率が改善した．

キーワード 音声認識，日本人英語，文法的な誤り

Speech recognition of English spoken by Japanese native speekers using N-gram trained from generated text

Ryohei TSUTSUI†, Motoyuki SUZUKI†, Akinori ITO†, and Shozo MAKINO†

† Graduate school of engineering, Tohoku University
6-6-5, Aoba, Aramaki-aza, Aoba-ku, Sendai, Miyagi, 980-8579 Japan
E-mail: †{tutui,moto,aito,makino}@makino.ecei.tohoku.ac.jp

Abstract Our goal is to develop a voice interactive CALL system which enables langage learners to practice words, phrases, and grammars interactively. In order to develop such a system, it is necessary to recognize learner's utterances correctly. We found that 4 or 5 states HMM works better than 3 states HMM in the case of recognition of English spoken by Japanese native speakers. Ngram language model trained from generated text achieves heigher speech recognition accuracy than FSA(Finite States Automata) language model.

Key words speech recognition, non-native English, grammatical error

1. はじめに

コンピュータを用いた外国語学習 (CALL, Computer Assisted Language Learning) システムとしては，従来，文字入力によるシステムが研究・開発されてきた．しかし，近年では“読み書き”の能力だけでなく，“聞く”・“話す”といった会話能力にも関心が高まっており，CALLシステムの分野でも，音声認識技術を利用して発音やイントネーションの練習を行うシステムなどが，盛んに研究されるようになってきた[1].

会話能力を向上させるには，発音の練習だけではなく，ロールプレイなどによって，新しく覚えた文法や表現を，実際に使う練習をする必要があると考えられている．そこで，学習者とシステムとの間で声を使ってロールプレイによって会話練習を行う，音声対話型のCALLシステムに関する研究も行なわれ始めている[2]～[4].

音声対話型CALLシステムでは，対話を円滑に進め，また，学習者に適切なフィードバックを行なうために，学習者の発話を発話した通りに認識する必要がある．しかし，これまでの研究では，外国語学習者の発話を高い精度で認識することはできていない．これは，外国語学習者が犯す発音上の誤りと，文法的な誤りのためである．

音響モデルについては，従来の研究では，日本人の英語を認識するときには，学習データに日本人の英語を使うといった検討などに限られ，HMMの構造などについての検討はほとんど行われてきていない．

また，言語モデルに関しては，Ehsani[2]は，タスクに関す

るテキストを大量に収集し，そこから Ngram を学習する方法をとった．また，阿部 [3] や Kweon [4] は，文法的な誤りを含む文を受理できるようなオートマトンをあらかじめ作成するという方法をとった．しかし，いずれも十分な認識率を得られていない．

そこで本研究では，音響モデルについては，HMM の構造について検討し，さらに，言語モデルに関しては，自由度の高い Ngram 言語モデルと，制約の強いオートマトン型言語モデルの中間的な方法として，Ngram 言語モデルと正解文との距離を用いた方法を提案する．

2. 音声対話型 CALL システム

音声対話型の CALL システムを用いて学習する際には，

（1） 重要な表現や文法事項を学習する．

（2） そこで学んだ表現などをコンピュータとの会話で実際に使ってみる．

（3） 間違いがあれば，学習者にフィードバックする．

という流れで進めていく．この様子を図 1 に示す．

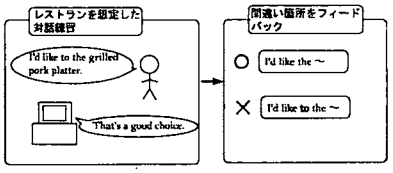

図 1 音声対話型外国語学習システム

3. 日本人英語用音響モデル

3.1 状態数と混合数

通常，音声認識において，音響モデルとして用いられる HMM の構造としては 3 状態 left-to-right モデルが用いられる [5]．このモデルは，ネイティブの発音モデルとしては適しているが，ノンネイティブの発音モデルとしては必ずしも適しているとはいえない．なぜなら，日本人が英語を話す時には，子音の後に母音が挿入されることが多いなど，英語では 1 つの音素で表現される音が実際には複数の音を含んでしまうことが多いためである．

この問題に対処するため，図 2 のように，直列に状態を増やしたモデルや，図 3 のように，途中から最終状態へ遷移するスキップありのモデルの検討を行なった．

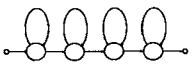

図 2 4 状態 HMM

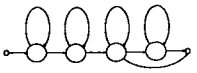

図 3 4 状態スキップあり HMM

3.2 音素認識実験

まず，HMM の状態数と，出力確率分布の混合数を変化させて音素認識実験を行なった．それ以外の実験条件を表 1 に示す．

音素定義は ERJ の方式に基づいた．また，サイレンスのモデルとして，3 状態 HMM を，そしてショートポーズモデルとしては 1 状態 HMM をそれぞれ用いた．

表 1 音素認識実験条件

特徴量	MFCC, Δ MFCC, Δ Δ MFCC, Δ pow, Δ Δ pow
学習データ	ERJ データベース [6] の 男性 100 人分 (約 10 万単語)
テストデータ	男性 9 人，計 17 発話
デコーダ	julian multipath 版 ver3.5.3

monophone での音素認識の結果は図 4 のようになった．

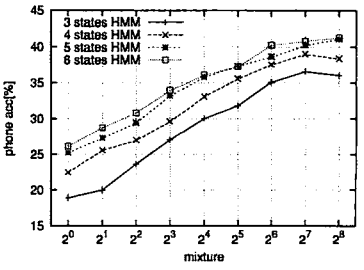

図 4 monophone での音素認識精度

直列に状態数を増やした HMM を用いることにより，3 状態 HMM を用いた場合に比べて，音素認識精度が向上するということが分かった．

また，スキップありのモデルに関しても，同様に monophone のモデルを学習し，音素認識実験を行なったが，3 状態 HMM とほぼ同じ音素認識率であった．

この原因としては，スキップの遷移確率が相対的に高くなり，結局 3 状態 HMM と同じモデルになってしまったためではないかと考えられる．

次に，triphone での音素認識結果を図 5 に示す．ここでは，木構造クラスタリングにより，3 状態 HMM は 300 状態まで状態を共有し，4 状態 HMM は 1500 状態まで状態を共有した．

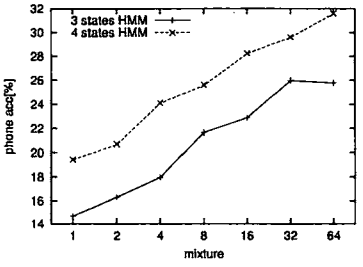

図 5 triphone モデルの音素認識精度

この図から分かるように，やはり 4 状態 HMM を用いることにより 3 状態 HMM に比べ，高い音素認識率が得られた．しかし，monophone モデルに比べ，低い音素認識率しか得られなかった．

この原因としては，ERJ データベースに登場する文のバリ

エーションが少なく，音素の組合せに偏りが生じてしまったことが考えられる．

4. オートマトン型認識文法を用いた音声認識

4.1 はじめに

従来の音声対話型CALLシステムでは，音声対話用の認識文法を手書きで記述するという方法がとられてきた[3]．しかし，人手で記述する方法では，システムを作る度に誤りを含む認識文法を記述し直さなければならないという問題がある．そこで，本研究では，統計的な方法により文法誤りを含んだ認識文法を自動生成し，音声認識に用いることを試みる．

なお，以降の実験では，以下のようにして収録したテストデータを用いる．

（1） 英語の会話文を覚える (1対話あたり数分)．

（2） 日本語訳を見ながら，コンピュータと対話する

ここで，最初に覚えた英語文を**正解文**と呼ぶ．

4.2 誤りルールを用いた認識文法の生成

正解文と**誤りルール**とを用いて，認識文法を以下のような手順で生成した．

（1） 正解文を受理するオートマトンを作成．

（2） オートマトンのすべての単語に対し，誤りルールを調べ，その単語に対する誤りルールがあれば，それと並列なパスを付加する．

（3） ステップ2を所望の回数繰り返す．

なお，ここで，“誤りルール”とは，“were を are に間違える”というような，正しい単語列と，それに対する誤った単語列の対の集合のことをいい，前後関係は考慮しない．

パス付加の一例として，“were を are に間違える”という誤りルールを1回だけ適用したときの様子を図6に示す．

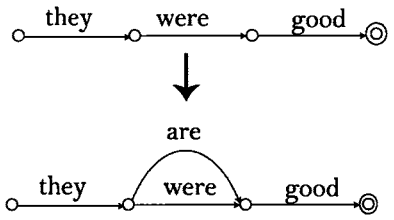

図6 パス付加の一例

4.3 誤りルール

誤りルールとしては，コーパスによる誤りルールと，一般的な誤りルールの2つを利用した．

コーパスによる誤りルール

コーパスによる誤りルールは，実際に日本人が話したデータから，単語ごとに，どの単語をどのように誤るかという情報を取り出したものである．

ここでは，誤りルールを得るためのコーパスとしてSSTコーパス[7]を用いた．このコーパスは日本人の話した英語を書き起こしたものであるが，実際の発話文だけでなく，誤っている箇所には本来どう言うべきであったかということが合わせて示されている．このデータから，どのような誤りが起こっているのかという情報を抽出する．

実際にSSTコーパスに含まれていた誤りの上位10個を表2に示す．

表2 SSTコーパスに含まれていた誤り (10位まで．またεは空の文字列を表す．)

順位	頻度	正解単語	誤り単語	誤りのタイプ
1	1227	a	ε	脱落
2	896	the	ε	脱落
3	403	ε	the	挿入
4	299	a	the	置換
5	207	to	ε	脱落
6	148	ε	a	挿入
7	139	is	ε	脱落
8	126	in	ε	脱落
9	118	the	a	置換
10	115	an	ε	脱落

表には挿入誤りも含まれているが，前後関係を考慮しないため，挿入誤りを誤りルールとして採用することはできない．すなわち，SSTコーパスに含まれていた誤りから，挿入誤りを除いたものを**コーパスによる誤りルール**として採用する．

一般的な誤りルール

コーパスによる誤りルールには，挿入誤りが含まれていない．しかし，表．2から分かるように，冠詞の挿入誤りは頻発している．そこで，名詞の前にはaまたはtheが挿入されるというルールを誤りルールに追加した．このように，個別に用意した誤りルールのことを**一般的な誤りルール**と呼ぶ．

なお，このときの品詞の同定にはBrill's Tagger[8]を用いている．

4.4 生成した文法の評価実験

第4.2節の方法により文法を生成し，そのカバー率を調べる実験を行なった．実験条件を表.3に示す．

表3 文法評価実験の条件

学習データ	SSTコーパスの約13万語
テストデータ	SSTコーパスの10人分，約6千語 正解文に対する 実際の発話文の誤り率 15.1%
繰返し回数	1回から2回

結果を図.7に示す．なお，図中のcは，コーパスによる誤りに，gは一般的な誤り (ここでは冠詞の挿入のみ) にそれぞれ対応していることを示す．

一般的な誤りとして，挿入誤りに対応することにより，1回の繰返しだけで，カバー率を，96%程度まで上げられることが分かった．

また，このときの，正解文に対するネットワークのPerplexityを図8に示す．なお，この際パスの遷移確率はすべて等しいとみなしている．

この結果から，一般的な誤りを用いると，カバー率は若干改善し，一方，Perplexityは，大きく増大してしまうということ

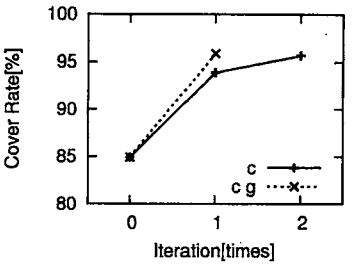

図 7 カバー率

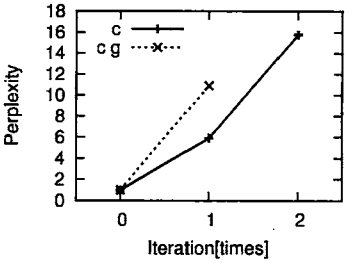

図 8 Perplexity

が分かる．したがって，一般的な誤りを用いた場合の認識率は，用いない場合に比べて，悪化してしまうと予想される．

4.5 音声認識実験

次に，第 4.1 節の方法で収録したデータを用いて，音声認識実験を行なった．

実験条件を表 4 に示す．なお，特徴量，学習データ，デコーダは表 1 に示したものと同じである．

表 4 音声認識実験条件

音響モデル	monophone
	512 混合 5 ループ HMM
テストデータ	男性 11 人，1 人当たり 3 文
	(正解文に対する誤り率：26.9%)

実験結果は，表 5 のようになり，一般的な誤りを使わない方が高い認識率を得ることができた．

表 5 単語認識精度 (Acc)

コーパスの誤り (c)	69.1%
コーパス+一般的誤り (c g)	62.2%

この実験において，認識結果文を確認したところ，非母語話者（日本人）ですらおかしいと思う表現がいくつも含まれていた．これは，誤り確率や，前後関係を考慮せずに誤りルールを適用したためであると考えられる．

5. 生成したテキストからの Ngram の学習

5.1 はじめに

前章のオートマトン認識文法を用いた方法では，誤り確率を考慮できないという問題や，想定外の誤りに弱いという問題点があった．

これらの問題を解決するため，本章では Ngram を用いた音声認識について検討する．

対話システムに Ngram を用いようとする試みはいくつかある [9] が，それらの多くは，大規模なテキストから学習した Ngram を対話用に適応するという方法をとっている．しかし，本研究で想定しているような外国語学習システムでは，学習者は，まず重要な表現や文法を覚えてから対話を行なうため，発話のバリエーションは他の対話システムに比べ，少ないと考えられる．

そこで，今回は，最初に覚えた正解文に似た文を十分に生成し，それをもとに Ngram を学習することを試みる．

5.2 Ngram の学習の概要

Ngram の学習の具体的な手順は，次の通りである．

（1） 正解文に誤りルールを適用して誤りを含んだ文を生成
（2） それを学習データとして，Ngram 頻度を学習
（3） 別の一般的な Ngram 頻度と混合
（4） back off Ngram を生成

この様子を図 9 に示す．

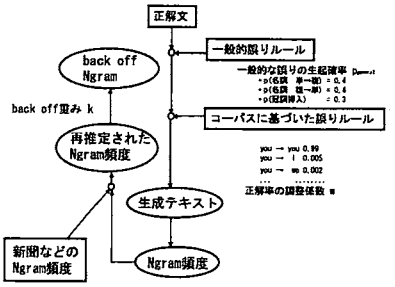

図 9 生成したテキストからの Ngram の学習

5.3 誤りルール

まず，誤りルールについて説明する．

ここで誤った文を生成する際に用いた誤りルールは，第 4.3 節で用いたものとは異なり，それぞれの誤りに対して，その誤りの生起確率が付与されたものである．たとえば，“was を 15%の確率で is と間違える” といったようになる．ただし，挿入誤りを誤りルールとして採用しないという点は，オートマトン文法を用いたときと同様である．

コーパスによる誤りルール

SST コーパスから誤りのパターンとその確率を合わせて取得し，それに誤り確率が大きくなるような補正を加えたものを**コーパスによる誤りルール**として採用する．補正を加えるのは，次のような事情があるためである．

SST コーパスは，自由な対話を書き起こしたものであるのに対し，現在検討しているシステムはあらかじめ正解文が決まっている．このため，SST コーパスから得た確率をそのまま用いると，誤り率が極端に低くなってしまう．たとえば，(a)“I like ...” という，表現と (b)“We like ...” という表現の両方が英語として正しい表現であったとする．このとき，SST コーパスにおいては，文法的に正しければ正解とみなすので，(a) も (b)

も正しいものとしてカウントされ，その結果，“I を we と誤る確率” や “we を I と誤る確率” は低くなってしまう (実際にはいずれも 1%以下)．一方，本システムでは，学習者は，日本語を見ながら英語を発話するため，文 (a) が正解文だとしても，文法的に正しい文 (b) を低くない確率で発話する可能性がある．このため，正解文 (a) に SST コーパスから得られた誤り確率をそのまま適用して誤り文を生成すると，(b) のような文がほとんど生成されず，テストセットパープレキシティが非常に高くなってしまうのである．

以上のような問題を防ぐため，実際には次のように誤り確率を補正して誤りルールとする．

まず，シグモイド関数を用いて，単語 s の正解率 $c(s)$ を

$$\hat{c}(s) = \frac{1 - e^{-\gamma w}}{1 + e^{-\gamma w}} \tag{1}$$

のように圧縮する．ここで，w は，圧縮重みであり，w が小さいほど正解率を低く補正できる．また，γ は，$w = 1.0$ のときに，$\hat{c}(s) = c(s)$ となるように正規化するための係数で，

$$\gamma = ln\frac{1 + c(s)}{1 - c(s)} \tag{2}$$

である．そして，単語 s を単語 t に間違える確率 $p(t|s)$ を，次のようにして補正する．

$$\hat{p}(t|s) = \frac{1 - \hat{c}(s)}{1 - c(s)} p(t|s) \tag{3}$$

一般的な誤りルール

正解文に品詞付けをし，

- 名詞の単数・複数
- 冠詞の挿入・脱落
- 形容詞の比較級変化

といった誤りを確率的に起こす．

この方法では，一般的な誤りの確率をどのように決定するかが問題となるが，予備実験の結果，確率が 0 でなければ，認識結果に大きな差はないことが分かった．

5.4 一般的な Ngram 頻度を用いた言語モデルの補正

正解文と誤りルールをもとに生成したテキストは，人為的なものであり，そこから学習した Ngram は偏っている場合がある．そこで，一般的なテキストの Ngram と混合することにより，その補正をすることを試みる．

5.4.1 語彙の決定と未知 Ngram の除去

まず，語彙については，生成テキストに含まれない語彙を未知語とする．さらに，生成テキストから学習した Ngram に含まれない Ngram を “未知 Ngram” と呼ぶ．

ここで，未知語や未知 Ngram の扱いを表 6 のように変えて認識実験を行なった．

表 6　語彙と未知語の扱い

名称	条件
mix simply	特になし
del unknown word	未知語を含む Ngram 頻度を 0 に
del unknown Ngram	未知 Ngram 頻度を 0 に

5.4.2 Ngram 頻度の混合

ある単語列 s の，一般的なコーパス (ここでは ANC [10]) における頻度を $N_a(s)$，生成テキストにおける頻度 $N_g(s)$ とし，それぞれの Ngram 頻度の和 S_a，S_g を

$$S_a = \sum_s N_a(s) \quad , \quad S_g = \sum_s N_g(s) \tag{4}$$

により定義する．

このときの Ngram 頻度の和 $N(s)$ を，混合重み w_r を用いて，

$$N(s) = \lceil 10 w_r \frac{S_g}{S_a} N_a(s) + 10(1 - w_r) N_g(s) \rceil \tag{5}$$

により求める．第 (5) 式で，全体を 10 倍しているのは，切り上げによる誤差を小さくするためである．

5.4.3 音声認識実験結果

音声認識実験の結果は図 10 のようになった．

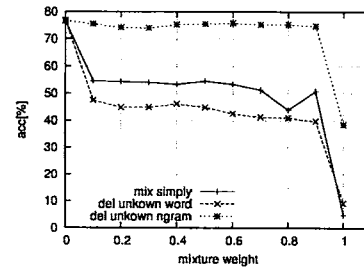

図 10　Ngram の混合

未知 Ngram を除去した場合，混合重みを大きくしても，認識率はあまり変化しないが，これは，混合重みを大きくすることにともなって，一般的なテキストの Ngram が強く働くようになり，正解文からすこし外れた文の認識精度が上がる一方，正解文に近い文のゆう度が下がってしまい，認識精度が悪化するというトレードオフによるものと考えられる．

6. 正解文との Levenshtein 距離を用いたスコア付け

6.1 正解文との距離を用いたスコア

前章の実験においては，学習データが限られた状況で Ngram 言語モデルを用いているので，平滑化がうまく機能せず，認識結果に，正解文からかけ離れた文が出力されるということもあった．

しかし，実際にはユーザは事前に文を覚えてから発話するので，覚えた正解文に近い文を話すと予想される．そこで，正解文に近い認識候補文のスコアを上げることで，認識精度を向上させることを試みる．

ここでは，認識結果文 $r = r_1, r_2, \ldots, r_M$ と，正解文 $s = s_1, s_2, \ldots, s_N$ との Levenshtein 距離 $d_L(r, s)$ を，スコアの算出に用いる．

具体的には，正解文が s である場面での認識候補文 r のスコアを

$$score(r|s) = (\text{認識時のスコア}) - \alpha \times d_L(r,s) \quad (6)$$

により求め，$score(r|s)$ を最大にするような r を最終結果とする．

なお，実際には，デコーダの Nbest 候補に対して上式を用いてリスコアリングを行ない，スコアを求める．

ここで，α は，リスコアリングの強さを決めるパラメータで，リスコアリング重みと呼ぶ．

6.2 実　　験

以上の方法により，リスコアリングを行ない，その際の認識率を調べた．

実験条件を Table 7 に示す．

表 7　実 験 条 件

Nbest	1000best
混合重み (w_r)	0.5(半々に混合),0(混合なし)

この結果は Fig. 11 のようになり，一般的な Ngram と混合した場合も，そうでない場合も，正解文との距離に対する重み α を適切に設定することで認識率が改善することが分かる．一方，Ngram を混合することによる，認識率の改善の効果は得られなかった．

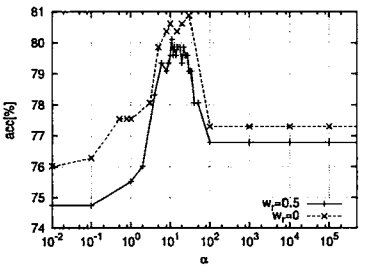

図 11　正解文との距離を用いたリスコアリング

さて，これまでは，単語認識精度で評価してきたが，実際に学習システムとして使用する際には，ユーザの誤りをどの程度適切に指摘できるかが重要となる．そこで，改めて，誤りの指摘の精度を誤りの指摘の再現率と適合率により評価したところ，図 12 のようになった．

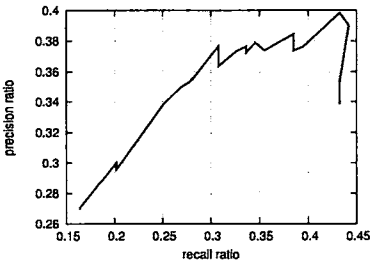

図 12　誤り指摘の再現率と適合率 ($w_r = 0.5$ のとき)

なお，この図において横軸は再現率 (recall ratio)，縦軸は適合率 (precision ratio) で，それぞれ

$$recall = \frac{\text{システムが指摘できた誤り数}}{\text{発話中の誤り数}} \quad (7)$$

$$precision = \frac{\text{システムが正しく指摘した誤り数}}{\text{システムが指摘した誤り数}} \quad (8)$$

により求めるが，いずれも誤りの種類と単語 ("I" を "you" に置換など) が完全に一致した場合のみ正解としている．

図 12 において，左下の部分は α が大きい場合に相当し，右下の部分は α が小さい場合に相当する．

やはり，誤り指摘の精度という観点から評価しても，α を適切に設定することで，その精度が改善しているといえる．

6.3 考　　察

正解文との距離に適切な重みを設定することで，認識率および，誤りの指摘の精度が改善した．これは，正解文から大きく外れた，極端に変な文が排除されたためであると考えられる．

また，認識精度約 80%に対して，誤りの指摘の精度が若干低いが，この原因としては，学習者が正しく発話した部分 (約 80%) は，ある程度の精度で認識できたものの，学習者が誤って発話した部分の認識率は低くなってしまったということがあげられる．

実際の認識結果には，日本人でさえ言わないであろうと思われる表現もいくつか含まれており，さらに検討が必要である．

7. ま と め

Ngram を用いた日本人英語音声認識について検討を行ない，正解文との距離を利用したスコアを用いることで，認識率が改善され，誤りの指摘の精度も向上することが分かった．

文　　献

[1] 中川聖一，牧野正三，壇辻正剛："音声言語処理技術を用いた語学学習システム"，日本音響学会誌，Vol59,No.6,pp.337-344(2003-06)

[2] Farzad Ehsani,Jared Bernstein,Amir Njmi: "An interactive dialog system for learning Japanese", Speech Communication 30, pp.167-177(2000)

[3] 阿部一彦，田中和世，河原達也，清水政明，壇辻正剛："対話型英語学習システムにおける日本人英語音声認識精度の検討"，音響講論，2-5-20，pp.113-114(2002-03)

[4] Oh-Pyo Kweon，Akinori ITO，Motoyuki SUZUKI，Shozo MAKINO："A Grammatical Error Detection Method for Dialogue-based CALL system"，Journal of Natural Language Processing，12，4，pp.137-156(2005)

[5] 鹿野清宏，伊藤克亘，河原達也，武田一哉，山本幹雄："音声認識システム"，オーム社 (2001)

[6] 峯松信明，富山義弘，吉本啓，清水克正，中川聖一，壇辻正剛，牧野正三："日本人話者による英語文・単語音声データベースの構築"，音響講論 1-Q-30(2001-10)

[7] 和泉絵美 他："日本人 1200 人の英語スピーキングコーパス"，アルク (2004)

[8] Eric Brill: "A Simple Rule-Based Part of Speech Tagger"，Proceedings of ANLP-92，3rd Conference on Applied Natural Language Processing，pp.152-155(1992)

[9] 伊藤彰則，好田正紀："対話音声認識のための事前タスク適応の検討"，信学技法 NLC96-50，SP96-81(1996)

[10] "American National Corpus"，http://www.americannationalcorpus.org/