単語の trigram を利用した文音声認識と自由発話認識への拡張

村上仁一　松永昭一

ATR 音声翻訳通信研究所

〒 619-02　京都府相楽郡精華町光台 2-2

あらまし

人間同士のコミュニケーションでは、「あのー」、「えーと」などに代表される冗長語や、言い淀みや言い誤りおよび言い直しなどが頻繁に出現する。このような音声でも認識できる、いわゆる自由発話の音声認識が、今後の重要な研究課題になると思われる。本論文では、初めに単語 trigram と Viterbi サーチ (one-pass DP) を基本とする文音声認識アルゴリズムを述べる。次にメモリ量や計算量を削減したアルゴリズムについて述べる。この改善は、ビームサーチや Viterbi の経路計算の改良などをすることにより得られた。この改良により、このアルゴリズムでは、各時刻・各状態において最尤の単語列を知ることができる。この特徴を生かして、音響モデルでは冗長語を認識しながら言語モデルでは冗長語をスキップすることにより、冗長語を含んだ音声を認識することができる。最後に自由発話の認識を行なった。その結果、このアルゴリズムの有効性が示された。

キーワード　• 2 重連鎖確率モデル　• one-pass DP　• 自由発話認識　• 冗長語

A Spontaneous Speech Recognition Algorithm with Pause and Filled Pause Procedure

Jin'ichi Murakami　Shouichi Matunaga

ATR Interpreting Telecommunications Research Laboratories

2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02

Abstract

This paper describes an effective recognition algorithm using word trigram models directly and a procedure of filled pauses in spontaneous speech. This recognition algorithm greatly reduces the memory requirements and computational costs by employing two techniques: beam search and an improved Viterbi search. With these methods, we can perform execution in a 15M byte space for about a 1500-word vocabulary. Next, focusing on spontaneous speech recognition, we look at a filled pause procedure to handle the many filled pauses in spontaneous speech. Even though the proposed method employs a simple procedure, we obtain a 64.4% sentence recognition rate for semi spontaneous speech and a 34.4% for spontaneous speech.

key words　• Word trigram model　• one-pass DP　• spontaneous speech recognition　• filled pauses

1　まえがき

従来研究されている音声認識システムでは、認識性能を向上させるために言語モデルを使用している [1]。経験的には、音節認識率と言語の perplexity と文認識率には相関があることが知られて、音声認識において使用される言語モデルは、基本的には、高いカバー率と低い perplexity が必要である。現在、音声認識に用いられる言語モデルとしては、簡潔さ・有効性などの点から単語の bigram モデルが主流である。しかし、単語の trigram は、確率を計算する方法や学習データ量どに多くの問題を含んでいるが一般的にはより小さな perplexity を示す。だが、trigram は、前の前の単語と前の単語が存在したときに現在の単語に遷移する確率であるため、認識アルゴリズムに trigram を組み込んだ場合、多くのメモリ量と計算量が必要になる。そのため、認識アルゴリズムとして 1 回の forward 探索で単語の trigram モデルを用いた例は見当たらない [2]。例えば BBN の文認識システムは、初めに forward 探索に bigram を使用して N-best リストを作製し、次に backward 探索で trigram を利用して再スコアをして最終的な認識結果を得ている。この方法では、正解の候補が forward 探索で N-best リストに入らない場合後の探索が無意味なものになり、trigram の有効性を損う可能性がある [3]。

この論文では、まず初めに trigram を利用した Vitebi サーチ (one-pass DP) の文音声認識のアルゴリズムについて述べる。次にメモリ量および計算量を削減したアルゴリズムについて述べる。これは、ビームサーチ法や Viterbi の経路計算の方法の改良などによって、計算量およびメモリ量を削減させた。そしてこのアルゴリズムを用いて朗読発話の文認識実験を行なった。

次にポーズがある音声でも、誤認識が起こり難いようにアルゴリズムを改良した結果について述べる。ポーズは音声データのあらゆる場所に出現する可能性がある。しかし言語モデルではこれに対応しきれないため、ポーズが原因で誤認識が起きやすい。ここで提案したアルゴリズムでは、各時刻・各状態において最尤の単語列を知ることができる。この特徴を生かして、音響モデルではポーズを認識しなから言語モデルではポーズをスキップして trigram の連鎖確率を計算することにより、ポーズがある音声でも誤認識が起こりにくくなる。

最後に、このアルゴリズムを自由発話の認識に適用した結果について述べる。自由発話では「あのー」、「えーと」に代表される冗長語を含む文が全体の約 5 割に達する。そしてこれらの冗長語は文の全ての場所に出現する可能性があるという点でポーズと似た性質がある。したがって冗長語の処理にはポーズ処理と同様な手法が使用できる。

2　単語の trigram model を使用した文音声認識システム

言語情報として単語の trigram を用いた場合の求める解は、文候補 $l(w_1, w_2, .., w_N)$ を以下のように定式化して、これを最大にする文 (単語列) $w_1, w_2, ..., w_N$ を選び出すことである。

$\sum \log(P_a(w_i)) + \alpha \times \sum \log(P(w_i|w_{i-2}, w_{i-1}))$　1)

ここで $P_a(w_i)$ は単語 w_i の音響尤度、$P(w_i|w_{i-2}, w_{i-1})$ は単語 w_{i-2} の次に単語 w_{i-1} が現れたときに w_i に遷移する確率、α は音響尤度と単語の尤度を結びつける結合定数である。

従来、連続単語認識のためのアルゴリズムとして知られている Viterbi サーチ (one-pass DP) は各単語の HMM の最後の状態と単語の最初の状態の遷移において trigram の確率を掛けることによって 1) 式を満たす最尤の単語列の候補を計算することができる。ただし、 trigram は 2 つ前の単語から現在の単語に遷移する確率値であるため、認識アルゴリズムでは、現在の単語と 1 つ前の単語の最大累積尤度を、つねに保持する必要がある。そのため bigram と比較すると、必要なメモリ量が大幅に増加する。認識単位を単語とした場合のアルゴリズムを表 2 に示す。

図 1 に、このアルゴリズムの簡略図を示す。この図では、認識語彙数を w_1 と w_2 の 2 単語で、単語の HMM は 4-state 3-loop で、状態は 0 から 2 までとする。縦軸は HMM の状態で、横軸は時間で、奥行きは語彙を示している。

図中○1から○7は表 1 に示される状態までの最大累積尤度であるとする。

表 1: 図中の状態

番号	時間	前の単語	現在の単語	現在の状態
○1	$t-1$	w_2	w_2	0
○2	$t-1$	w_2	w_2	1
○3	t	w_2	w_2	1
○4	$t-1$	w_2	w_2	2
○5	$t-1$	w_1	w_2	2
○6	$t-1$	w_2	w_1	0
○7	t	w_2	w_1	0

単語の最初の状態以外は、前時刻の同一状態と前時刻の 1 つ前の最大累積尤度の 2 遷移のうち、最大累積尤度の高い方を選択する。例えば、○3は○1の遷移と○2から遷移の最大累積尤度の高い方を選択する。しかし、単語の最初の状態は、前時刻の最初の同一の最大累積尤度と各認識単語の最後の最大累積尤度に現在の単語に遷移する trigram の連鎖確率値を掛けたものから遷移の最大累積尤度の高い方を選択する。例えば、○7は○4に trigram の値 $(p(w_1|w_2,w_2)^\alpha)$ を掛けたものと○5に trigram の値 $(p(w_1|w_1,w_2)^\alpha)$ と○6の遷移の尤度の高い方を選択する。これを全状態に対して計算を行なう。

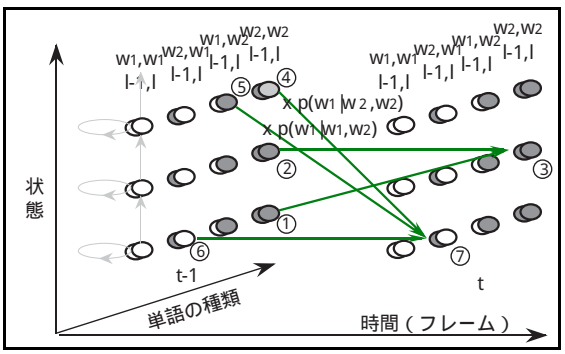

図 1: 単語の trigram を用いた Viterbi サーチの略図

表 2: 単語の trigram を用いた Viterbi サーチのアルゴリズム

[定義]

l_w : 単語 w における状態数
a_{ij}^w : 単語 w における状態 s_i から状態 s_j への遷移確率
$b_j^w(v)$: 単語 w の状態 s_j におけるベクトル v の出力確率
$P(w_0|(w_2,w_1))$ 単語 w_2,w_1 が出現したときに w_0 に遷移する確率
Q : 語彙数
T : 入力フレーム数
$O(t)$: フレーム t における観測ベクトル
$G_t(w_1,w_0,i)$: 前単語 w_1, 単語 w_0, 状態 i でのフレーム t までの最大累積尤度
α : 音響尤度と言語の連鎖確率の結合値

[初期化]

$w_0 = 0,...,Q-1$ において step1 を実行
1) $G_0(start,w_0,0) = P(w_0|start,start)^\alpha$
$start$ は文頭を意味

[Viterbi サーチ]

$t = 0,1,..,T-1$ において step2,step6 を実行
2) $w_1 = 0,...,Q-1$ において step3 を実行
3) $w_0 = 0,...,Q-1$ において step4 を実行
4) $i = 0,1,...,l_{w_0}-2$ において step5 を実行
5) $G_t(w_1,w_0,i) =$
$\max(G_{t-1}(w_1,w_0,i) \times a_{i,i}^{w_0} \times b_i^{w_0}(O_t),$
$G_{t-1}(w_1,w_0,i-1) \times a_{i-1,i}^{w_0} \times b_{i-1}^{w_0}(O_t))$

[単語境界の計算]

6) $w_1 = 0,1,...,Q-1$ において step7 を実行
7) $w_0 = 0,1,...,Q-1$ において step8 を実行
8) $\Delta = \max_{0 \le w_2 \le Q-1}(G_{t-1}(w_2,w_1,l_{w_1}-2)$
$\times a_{l_{w_1}-2,l_{w_1}-1}^{w_1} \times b_{l_{w_1}-1}^{w_1}(O_t) \times P(w_0|w_2,w_1)^\alpha$
もし $\Delta \ge G_t(w_1,w_0,0)$ ならば $G_t(w_1,w_0,0) = \Delta$

3 計算量およびメモリ量を削減した文音声認識アルゴリズム

表 2 に示したアルゴリズムを実際のコンピュータにインプリメントしても、大量のメモリが必要であるため小語彙の認識システムしか作成できない。そこでメモリ量と計算量を削減するために、次に述べるようにアルゴリズムを改良した。このアルゴリズムは HP735 で語彙数 1500 のとき、メモリ量 15Mbyte 平均文認識時間平均 1 分 30 秒 (リアルタイムの約 50 倍) で動作した。

3.1 ビームサーチ

Viterbi サーチは、全ての尤度の計算が終了した後に、累積尤度が最大の単語列を選択するアルゴリズムである。したがって各フレームごとの尤度計算において累積尤度の低い単語列は、以後の探索か

ら除外できる可能性が高い。そこで、フレームごとに最尤なものからある個数 (ビーム幅 b) のみ計算を続けることにより、計算量およびメモリ量が大幅に減らせる [4]。具体的には、すべての w_1, w_0, i に対して 表 2 step5 の式の計算のかわりに、最も高い累積尤度から、ある個数 (ビーム幅 b) のみを計算する。したがって必要なメモリ量は $G_t(w_1, w_0, i)$ を記憶するために、フルサーチは [認識語彙数 $^2\times$ 単語の状態数] が必要であるのに対し、ビームサーチでは [ビーム幅 b] しか必要としないため大幅に削減できる。同様な比率で計算量も削減できる。

3.2 ビームの絞り方

ビームの絞り方には、次の 2 つの方法がある。

1. 尤度の閾値値で、計算を打ち切る方法
2. 一定の個数を残す方法

尤度の閾値で計算を打ち切る方法は、計算量が少なくてすむためよく利用される。しかし、認識を行なう前に予め閾値を決めておく必要があるため、認識の途中で全ての経路が打ち切られたり、ビーム幅が急激に広がったりすることがある。一定の個数を残す方法は、フレームごとにソーティングが必要になるため、これが計算量の増大を招くと考えられてきた。しかし、フレームごとに上位からビーム幅の個数を示す尤度だけを算出することによって、同様な結果を得ることができる。この方法を採用することによって通常のソーティングに比較すると計算量が大幅に削減できる。具体的には N 個のデータがあったとき、フルソートでは $O(N \log_2 N)$ の計算量が必要であるが、上位からビーム幅の個数を示す尤度を算出すれば計算量は $O(log_2 N)$ ですむ。このアルゴリズムでは、後者を選択した。

3.3 Viterbi サーチの経路計算

Viterbi サーチでは、最尤の単語列の結果を得るために 2 つの方法が考えられる。

1. トレースバック

   各時刻・各状態において、最大累積尤度を計算したときに、選択した経路を記憶しておく。そして尤度の計算が終了した後トレースバックを行ない最尤の単語列を得る [1]。この方法は、各時刻・各状態において、選択した経路を記憶するために [ビーム幅 b $\times$ 最大認識時間のフレーム長] のメモリ量が必要である。しかし計算量は少なくて済む。

2. 最大累積尤度と同時に計算

   各時刻・各状態において、最大累積尤度の計算と同時に、選択した経路を次の状態に渡す。この方法ではトレースバックをしないで最尤の単語列を得ることができる。また、必要なメモリ量は [ビーム幅 b $\times$ 文を構成する最大単語数] であるが、一般的には文を構成する最大単語数は最大認識時間のフレーム長より少なくて済むため、トレースバックを行う方法より必要なメモリは減少する。そのかわり、必要な計算量はやや増加する。

前者は、計算量が少なくて済むため良く利用されている。後者は、前者と比較すると計算量が増加するが、各時刻・各状態においてトレースバックをしなくても経路を知ることが可能であるため、言語モデルにおける left-right 型のパーザーと組み合わせることが容易である。また、多くの場合、前者と比較して小量のメモリですむ。このアルゴリズムでは、メモリ量の増加を押えるため、後者を選択した。

4 trigram model を使用した文認識システムの実験

4.1 実験条件

単語の trigram を用いた文認識システムの認識性能を把握するために認識実験を行なった。実験は特定話者認識および不特定話者認識の 2 つの様式で行なった。HMM の学習データには、特定話者認識の場合はテストデータと同一話者の 2670 単語発声を使用し、不特定話者認識の場合は男性話者 12 名の 736 単語発声を利用した。テストデータは国際会議の問い合わせの文 (通称モデル会話) で、話者はナレータである。その他の実験条件を表 3 に示す。なお、音声データの前後には約 200ms のポーズが付加されている。実験文数は 261 文である。また、trigram の連鎖確率値は、ＡＴＲの対話データベースのなかから国際会議の予約に関するデータ約 1 万 2 千文章、約 17 万単語にテストデータのテキストを加えて計算した。

表 3: 文音声認識の実験条件

基本アルゴリズム	Continuous mixture HMM + Beam search + word trigram
Mixture 数	最大 14 (各音素によって変化)
1 音素あたりの状態数	4-state 3-loop left-right model
使用パラメータ	LPC ケプストラム 16 次 + パワー + Δ パワー+Δ ケプストラム 16 次
ウインド幅	20ms
フレーム周期	5ms
HMM の学習音声	
(特定話者認識)	テストデータと同一話者の 2670 単語発声
(不特定話者認識)	男性話者 12 名の 736 単語発声
音素カテゴリ数	52 音素
認識単語数	1567
ビーム幅	4096
duration control	なし
言語情報	単語の trigram
認識単位	文
実験文数	261 文
発声様式	朗読発話
発声内容	国際会議の問い合わせ (通称モデル会話)
trigram の連鎖確率の推定に使用したテキストデータ量	約 1 万 2 千文章 171978 単語

4.2 実験結果

実験結果を表 4 に示す。なお比較のために言語モデルに bigram を使用した時の実験結果も示した。実験の結果、言語情報として trigram を用いるとき、特定話者認識においては文認識率で 66.7%、不特定話者認識では 28.0%が得られた。そして言語モデルとして trigram を使用した場合、bigram よりもかなり高い認識性能が得られることがわかった

表 4: 認識実験の結果　認識率 (%)

model	特定話者認識	不特定話者認識
bigram	33.3% (87/261)	0.0% (0/261)
trigram	66.7% (174/261)	28.0% (73/261)

text-closed ビーム幅:4096 α:1

実験結果は文認識率で計算した。したがって認識結果が 1 文字でも異なった場合は誤認識としている。誤認識の結果を見ると、「会議に申し込みたいのですが」を「会議に申し込たいんですが」のような助詞の「の」を「ん」とするの誤りが多かった。このような意味的に正しい文を正解に含めた場合、認識性能は文認識率で約 80%、不特定話者認識の場合は約 40%であった。

4.3 ポーズのスキップ

ポーズは、文節と文節の間に出現することが多いが、音声データのあらゆる場所に出現する可能性がある。しかし、言語モデルでこれに対応しきれないため、ポーズの区間で誤認識が起きやすい。そこで単語と単語の境界にポーズがあっても、誤認識が起き難いようにアルゴリズムを改良した。

ここで提案したアルゴリズムでは、各時刻・各状態において最尤の単語列を知ることができる。そこでポーズを 1 単語と考えて、ポーズに接続される trigram の値は 1.0 にする。そしてポーズ以外の単語に接続されるときポーズをスキップして trigram を計算する。例えば「東京都 港区 新橋 /*pause*/ 1 丁目」 では P (新橋 | 東京都 港区) $\times 1.0 \times$ P(1 丁目 | 港区 新橋) と計算する。このようにすると、近似解ではあるが、ポーズをスキップして単語 trigram を用いたときの最尤の解が得られる。

アルゴリズムでは、表 2 step 8 を表 5 のように変更する。

表 5: 改良したアルゴリズム (ポーズのスキップ)

8)
if $w_0 =$/pause/
$\Delta = \max_{0 \le w_2 \le Q-1}(G_{t-1}(w_2, w_1, l_{w_1} - 2)$
$\times a^{w_1}_{l_{w_1}-2, l_{w_1}-1} \times b^{w_1}_{l_{w_1}-1}(O_t)) \times 1.0$
else if $w_1 =$/pause/
$\Delta = \max_{0 \le w_2 \le Q-1}(G_{t-1}(w_2, w_1, l_{w_1} - 2)$
$\times a^{w_1}_{l_{w_1}-2, l_{w_1}-1} \times b^{w_1}_{l_{w_1}-1}(O_t) \times P(w_0|w_3, w_2)^{\alpha})$
else
$\Delta = \max_{0 \le w_2 \le Q-1}(G_{t-1}(w_2, w_1, l_{w_1} - 2)$
$\times a^{w_1}_{l_{w_1}-2, l_{w_1}-1} \times b^{w_1}_{l_{w_1}-1}(O_t) \times P(w_0|w_2, w_1)^{\alpha})$
if $\Delta \ge G_t(w_1, w_0, 0)$ then $G_t(w_1, w_0, 0) = \Delta$

このように改良したアルゴリズムを用いて認識実験を行なった。実験条件は表 3 と同一である。この結果を表 6 に載せる。この結果からポーズのスキップをすることにより、認識性能が向上することがわかる。

表 6: 認識実験結果 (ポーズのスキップ) 認識率 (%)

model	特定話者認識	不特定話者認識
bigram	49.4% (129/261)	31.4% (82/261)
trigram	71.6% (187/261)	61.7% (161/261)

text-closed ビーム幅:4096 α:1

4.4 ポーズの学習

不特定話者認識の実験において誤認識された結果を調べると、音声データの最初のポーズ区間から

誤認識していることがわかった。そこでテストデータの先頭の無音区間を利用して、ポーズの HMM を再学習して、認識実験を行なった。このときの実験結果を表 7 に載せる。これからわかるように認識性能がさらに向上した。そして不特定話者認識と特定話者認識の認識率の差は、かなり小さくなった。また特定話者認識において意味的に正しい文も正解に含めたとき、認識率は 99%に達した。

表 7: 認識実験の結果 (ポーズのスキップ、ポーズ学習)　認識率 (%)

model	特定話者認識	不特定話者認識
bigram	60.5% (158/261)	44.8% (117/261)
trigram	88.9% (232/261)	83.9% (219/261)

text-closed ビーム幅:4096 α:1

4.5　ビーム幅

ビームサーチは、最終的な累積尤度が最大になる単語単語列は、各フレーム時においても累積尤度は高いとの仮定に成り立っている。ここではビーム幅を変えた時の文認識率の変化を調べた。ビーム幅以外の実験条件は、表 3 と同一である。また、4.3 節および 4.4 節で述べたポーズ処理はおこなっている。

この実験結果を図 2 に示す。これからビーム幅を広げるにしたがい認識性能が向上するが、ビーム幅が 1024 を越えると、認識性能はあまり向上しないことがわかる。

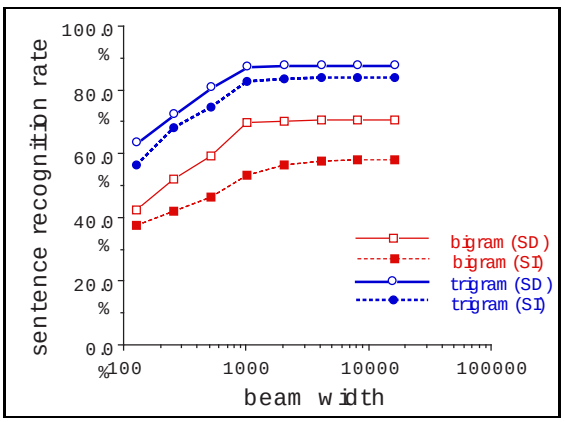

図 2: ビーム幅を変化させたときの変化　認識率 (%)
text-closed α:1

4.6　音響尤度と言語の連鎖確率の結合値

これまでの実験では、音響尤度と言語の連鎖確率の結合値 α を 1 とした。ここではこの結合値を変えた時の文認識率の変化を調べた。音響尤度と言語の連鎖確率の結合値以外の実験条件は、表 3 と同一である。また、 4.3 節および 4.4 節で述べたポーズ処理はおこなっている。

この結果を図 3 に示す。この実験から音響尤度と言語の連鎖確率の結合値 α が 16 のとき最も高い文認識率を示した。

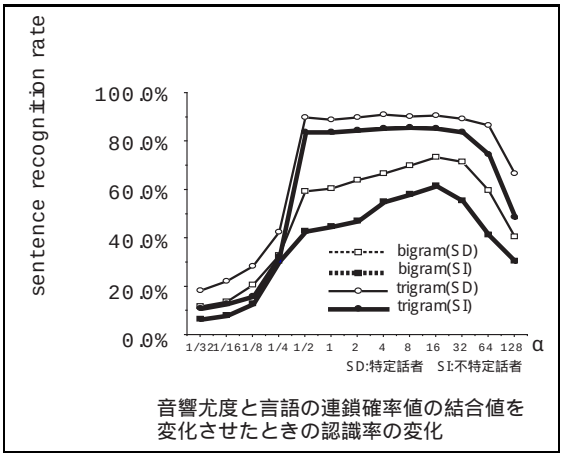

図 3: 音響尤度と言語の連鎖確率の結合値を変えたときの認識性能の変化　認識率 (%)
text-closed ビーム幅:4096

4.7　text-open data における認識率

trigram の連鎖確率の計算に使用するテキストデータの学習量に対する文認識率の変化を調べた。実験は、言語モデルとして bigram と trigram、特定話者認識と不特定話者認識、さらに text-close data (ＡＴＲの対話データベースにテストデータを加えて連鎖確率を計算した場合) と text-open data (Ａ

ＴＲの対話データベースから連鎖確率を計算した場合) の合計 8 種類行なった。実験条件は、表 3 と同一である。また 4.3 節および 4.4 節で述べたポーズ処理はしている。また音響尤度と言語の連鎖確率の結合値 α は 32 とした。

この実験結果を図 4 に示す。この図では横軸は trigram の連鎖確率値を計算するのに使用した学習データの単語数で縦軸は文認識率である。この実験では、text-closed data では trigram のほうが bigram と比較してかなり高い認識性能が得られるが、 text-open における実験では、bigram のほうが trigram よりも認識性能は高いことがわかる。

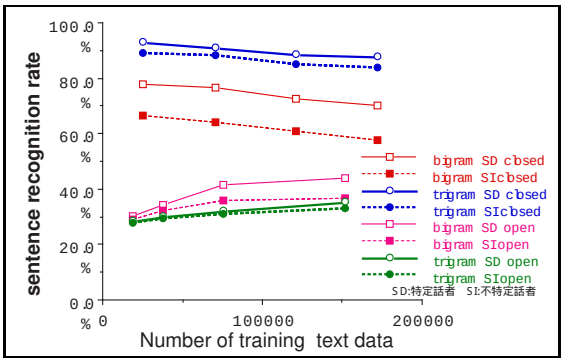

図 4: 学習データ量における認識結果の変化　認識率 (%)

5　自由発話の文認識実験

従来の音声認識システムの多くは丁寧に発声された音声を入力対象にしている。しかし、人間同士のコミュニケーションでは、「あのー」「えーと」などに代表される冗長語や、言い淀みや言い誤りおよび言い直しなどが頻繁に出現する。このような音声でも認識できる、いわゆる自由発話の音声認識が、今後の重要な研究課題になると思われる。自由発話に特有な言語現象のなかで、冗長語は対話文の約 5 割に出現する [5]。したがって冗長語を含む音声の認識が、自由発話音声認識の第一歩と考えられる。ここでは、冗長語を含む音声でも誤認識を起こしにくいアルゴリズムを考えた。

5.1　冗長語の処理

「あのー」、「えーと」に代表される冗長語は文の全ての場所に出現する可能性があるという点でポーズと似た性質がある。したがって冗長語の処理はポーズ処理と同様な手法が使用できる。つまり、音響モデルでは冗長語を認識しながら、言語モデルでは冗長語をスキップする。例えば「東京都 港区 新橋 えーと 1 丁目」 のとき $P($ 新橋 $|$ 東京都 港区 $) \times 1.0 \times P($1 丁目 $|$ 港区 新橋 $)$ と計算する。

5.2　自由発話の音声データ

ここでは、冗長語の処理の有効性を自由発話の音声で調べた。自由発話の定義は人によって異なるが、本論文では、話者がテキストを見ないで発声した音声を自由発話と定義した。そして以下に示すような方法で収録した音声データを実験に使用した。

1. 朗読発声
   テキストを読みあげた音声データ。冗長語や言い淀み・言い直しは無い。

2. 疑似自由発話
   冗長語が書かれているテキストを読みあげた音声データ。冗長語を除いて、「1 朗読発声」と発話内容は同一。言い淀み・言い直しは無い。

3. 自由発話
   話者はテキストを覚えて、その意図を理解し、自由に発話した音声データ。発話内容は「1 朗読発声」と異なる。言い淀み・言い直しは無い。

なお、話者は一般人である。

5.3　自由発話の文認識実験

認識実験は、不特定話者認識の trigram の場合のみ行なった。冗長語は「あのー」「えーと」「まあ」などを含めて 109 種類、定義した。また 4.3 節および 4.4 節で述べたポーズの処理は行なっている。その他の実験条件は表 3 と同様である。この実験結果を表 8 に示す。

実験の結果、冗長語を付けた疑似自由発話の音声では 64.4%、自由発話音声では 34.4%の文認識率が得られた。また、冗長語の処理をしたほうが、疑似自由発話でも自由発話でも、ポーズの処理のみの方法より有効であること、および朗読発声の文認識においても従来の方法と比較して認識率が

あまり低下しないこと (82.6% → 74.8%) を考慮すると、このアルゴリズムは自由発話の認識において有効であると言える。

表 8: 自由発話の文認識実験結果

文認識率 (%)

発話様式	+ポーズ処理	+ポーズ処理+冗長語処理
朗読発声	82.6%	74.8%
疑似自由発話	26.7%	64.4%
自由発話	14.1%	34.4%

6 考察

6.1 ポーズ処理

今回の実験から、誤認識の原因になっている音声に含まれるポーズの対策として、言語モデルではポーズのスキップ、音響モデルではポーズの HMM を学習することで文認識性能が向上することが示された。したがって音声認識性能を向上させる上で、ポーズは促音やクロージャとも併せて、別の方法も考慮する必要がある。特にポーズの HMM の学習に関しては、混合分布の平均値を移動させるアダプテーションの方法をとるほうが好ましいと考えている。

6.2 ビーム幅

ビーム幅は語彙数と正の相関を持つと考えられる。しかし実験ではビーム幅が 1024 を越えると、認識性能はあまり向上しないことが示された。認識語彙数を変化させた実験を行っていないため明確ではないが、このビーム幅 1024 は語彙数 1567 に近いことから、ビーム幅は語彙数程度で十分であると思われる。ただし、ここで実験に用いた話者はナレータであるため、音声は非常にクリーンに発話されている。したがって、通常の話者の音声ではこのビーム幅では不足する可能性もある。

6.3 音響尤度と言語の連鎖確率の結合値

音響尤度と言語の連鎖確率の結合値を変化させた時の文認識率の変化を調べた実験から α が 16 のとき最も高い文認識性能が得られた。

しかし、単語の HMM と単語の bigram を考えて、これらを組み合わせたモデルは ergodicHMM に似たモデルになる。そして単語の bigram の値は 1 つの単語の HMM の最終状態の遷移確率を別の単語に接続されたときの値の分配率になる。この時の音響尤度と言語の連鎖確率の結合値 α は 1 になる。この値は trigram でも同様であると考えられる。したがって理論的には音響尤度と言語の連鎖確率の結合値 α は 1 であると思われる。

6.4 学習データと text-open data の認識率

今回の実験で用いたテキストデータベースのデータ量に対するエントロピーと “頻度別出現率” の変化を調査した。 なお “頻度別出現率” は次のように定義している。

“頻度別出現率 60%” が示す値は、学習データの中で 60%をカバーするのに必要な最小のマルコフ連鎖確率の種類の数である。また “頻度別出現率 100%” が示す値は、学習データ量全てをカバーするのに必要なマルコフ連鎖の種類の数である。

調査は頻度別出現率 60%、頻度別出現率 80%、頻度別出現率 100%、およびエントロピーの合計 4 つの値で行なった。この結果を図 5 に示す。この図から、データ量が増加するにしたがい、エントロピーも増加していることがわかる。またデータベースの語彙の 58.8%(3486/5933) は 1 回しか出現していなかった。また、単語 trigram の全ての組合せの中の 77.9%(60847/78138) は 1 回しか出現していなかった。これらの結果はデータ量の不足を示している。そのため、trigram の値の信頼性が低く、 text-open data と text-closed data の認識性能に大きな差が生じたと考えている。

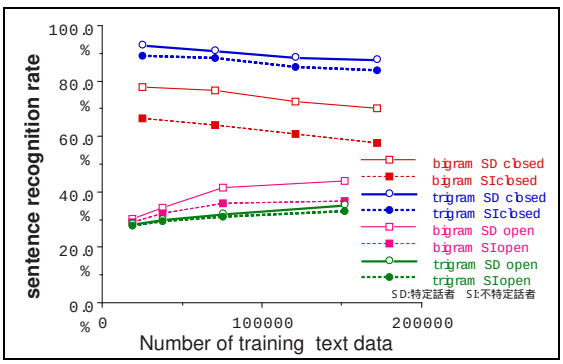

図 5: 学習データの入力データに対するエントロピーおよび頻度別出現率の変化

6.5 自由発話認識の問題点

実験で使用したテキストデータは小規模であるため、trigram の連鎖確率値の信頼性が低いと思われる。そのため、今回の実験では text-open data の認識性能が低くなったと考えている。また text-open data では未知語の出現が避けられない。したがって、今後の自由発話認識の課題として、text-open data に対する認識性能の向上と共に未知語処理が挙げられる。

7 まとめ

本論文では、初めに単語 trigram と Viterbi サーチを基本とする文音声認識アルゴリズムを述べた。次に計算量およびメモリ量を削減したアルゴリズムを提案し、その実験結果を報告した。実験の結果、朗読発話の不特定話者認識の text-closed data において、28.0%の文認識率が得られた。

次にポーズを考慮したアルゴリズムについて報告した。ポーズは音声データのあらゆる場所に出現する可能性がある。しかし、言語モデルではこれに対応しきれないため、ポーズの区間で誤認識が起きやすい。この改良されたアルゴリズムでは、各時刻・各状態において最尤の単語列を知ることができる。この特徴を生かして、音響モデルではポーズを認識しながら言語モデルではポーズをスキップすることにより、ポーズによる誤認識を少なくすることができる。また、テストデータの先頭の無音区間を利用して、ポーズの HMM を再学習した。これらのポーズの対策をすることにより 83.9%の文認識率が得られた。

最後に自由発話の認識を行なった。自由発話に特有な冗長語は音声のあらゆる場所に出現する可能性があるという点でポーズと似た性質がある。そこでポーズと同様な処理をすることにより、冗長語がある音声データでも認識が可能になる。自由発話の認識実験では冗長語処理をすることにより文認識率が 14.1%から 34.4%に向上し、このアルゴリズムの有効性が示された。

謝辞

研究の機会を与えて頂いた音声翻訳通信研究所 山崎泰弘社長および第一研究室 匂坂室長、そして熱心な御討論と有益な御助言をいただいた各研究室の皆様に感謝致します。

参考文献

[1] Kai-Fu Lee, “Large-Vovabulary Speaker Independent Continuous Speech Recognitton: The SPHINX System,” 15213 CMU-CS-88-148 (April 18, 1988).

[2] A.Averbuch, “An IBM PC Based Large-Vocabulary Isolated-Utterance Speech Recognizer,” Vol.1, 2.4.1 PP.53-56 ICASSP (1986).

[3] Rechard schwarts, et. “Comparative experiments on large vocabulary speech recongition”,ARPA Human Langauge Technology Workshop, (Mar. 1993)

[4] 迫江博昭, 藤井浩美, 吉田和永, 亘理誠夫, “フレーム同期化、ビームサーチ、ベクトル量子化の統合による DP マッチングの高速化,” 信学論 D,Vol.J71-D, No.9, pp.1650-1659(Sep. 1988)

[5] 村上 仁一、他: “自由発話音声認識における音響的および言語的な問題点の検討,” 信学技報, SP91-100, pp.71-78 (12 1991).