TR-IT-0031

# 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
# A Spontaneous Speech Recognition Algorithm with Pause and Filled Pause Procedure

村上仁一
Jin'ichi Murakami

1993.12

## 概要

本論文では、始めに単語 trigram と one-pass DP を基本とし、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って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、アルゴリズムの有効性が示された。

# 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

# A Spontaneous Speech Recognition Algorithm with Pause and Filled Pause Procedure

村上仁一
Jin'ichi Murakami
ATR 音声翻訳通信研究所
ATR Interpreting Telecommunications Research Labs.

## 概要

近年、文音声認識の研究が盛んに行なわれ、いくつかの研究機関で文音声システムが構築されている [4][2]。これらの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。本論文では、初めに単語 trigram と one-pass DP を基本とする文音声認識アルゴリズムを述べる。しかし、このアルゴリズムでは大量のメモリが必要である。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。最後にこのアルゴリズムを利用して自由発話の認識を行なった。その結果、アルゴリズムの有効性が示された。

## 1 まえがき

人とコンピュータのインターフェースとして、あるいは異なる母国語を話す人と人とのインターフェースとして、自然言語を認識できる文音声認識システムが研究されている [4][2]。これらの音声認識システムでは音響処理だけでは高い認識性能が得られないため、言語情報も利用している。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、中川等は、この関係を推定している [1]。この結果を見ると、言語の perplexity を下げると文認識率は著しく向上しているようだ。したがって、音声認識において使用される言語モデルには、基本的には、高いカバー率と低い perplexity が必要であると言えよう。

言語の perplexity を下げる方法には、大きくわけて 2 つの方法がある。1 つは、より複雑なルールを書くことで、もう 1 つは、確率的な言語モデルを採用することである。ルールに基づく言語モデルとしては、ネットワーク文法や文脈自由文法、単一化文法などが知られている。しかし、一般的に作られた文法の perplexity は高くなりがちなため、これを減少させるために、ルールを追加する必要がある。一方、カバー率をあげるためにも、ルールを書く必要がでてくる。したがって、ルールに基づく言語モデルは、一般的にルールの作成およびメインテナンスに、大きな負荷がかかる。確率的な言語モデルとしては、単純な bigram や trigram から、確率つきネットワーク文法や確率つき文脈自由文法などが提案されている。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 が現在のところ主流であると言える。しかし、言語モデルの perplexity から見ると、確率を計算する方法や、学習データ量、そして text-open data と text-closed data の差などの問題があるが、単語の trigram がもっとも小さい値を示しているようだ。

しかし、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を逐次計算する必要があるため、大量のメモリ量と計算量が必要である。そのため、文音声認識システムに、

直接単語のtrigramモデルを使用した例は見当たらない。IBMの認識システムは単語trigramをもちいているが、単語孤立発声の音声認識である[3]。鹿野[10]や南他[11]は、カテゴリと組み合わた、疑似的なtrigramを用いている。しかし、カテゴリを用いた場合、perplexityが増大するため、非文が出現しやすくなる。山田[6]らは、LRパーザーと組み合わせることによって、trigramのサーチ空間を減少させているが、この方法では、LRパーザも記述する必要があるため、言語モデルはかなり複雑になる。村上[8]らの研究は、音節のラティス構造を仮定したシミュレーションである。BBNやCMUの文認識システムは、始めにbigramを使用してN-bestリストを作製し、つぎにtrigramを利用して再スコアをしている。この方法では、正解の候補がN-bestに入らない場合があり、trigramの有効性が損なわれている恐れがある。

この論文では、まず初めにtrigramを利用したone-pass DPの文音声認識のアルゴリズムについて述べる。しかし、この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になるため、数十単語程度の小語彙認識しかできない。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。ここでは、ビームサーチやViterbiの経路計算の方法やtrigramの値の記憶方法を改良することなどによって、計算量およびメモリ量を削減させた。次に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。この実験の結果、単語のtrigramはbigramよりも高い認識性能が得られることが示された。

ところで、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、言語モデルではカバーしきれないため、ポーズの区間で誤認識が起きやすい。そこで全ての単語と単語の境界にポーズが入力されても文認識が可能なようにアルゴリズムを改良した。このアルゴリズムの改良によって、認識性能が向上した。

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。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[9]。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。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なると考えられる。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ようにアルゴリズムを改良することにより、冗長語を含んだ音声を認識することができる。この改良したアルゴリズムを用いて、自由発話の認識を行なった。この実験結果について報告する。

## 2 単語のtrigram modelを使用した文音声認識システム

この章では単語のtrigram modelを利用した文音声認識システムのアルゴリズムについて述べる。

言語情報として単語のtrigramを用いた場合、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$w_1, w_2, ..., w_N$を選び出すことである。

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

ここで$P_a(w_i)$は単語$w_i$の音響尤度、$p(w_{i+2}|w_i, w_{i+1})$は単語$w_i$の次に単語$w_{i+1}$が現れたときに$w_{i+2}$に遷移する確率、$\alpha$は音響尤度と言語の連鎖確率を結びつける結合定数である。

表1: 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<br>$a_{ij}^w$：単語$w$における状態$s_i$から状態$s_j$への遷移確率<br>$b_j^w(v)$：単語$w$の状態$s_j$におけるベクトル$v$の出力確率<br>$P(w_0\|(w_2, w_1))$ 単語$w_2, w_1$が出現したときに$w_0$に遷移する確率<br>$Q$：語彙数<br>$T$：入力フレーム数<br>$O(t)$：フレーム$t$における観測ベクトル<br>$G_t(w_1, w_0, i)$：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<br>$\alpha$ ：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<br>1) $G_0(start, w_0, 0) = P(w_0\|start, start)^\alpha$<br>$start$は文頭を意味 |
| [ Viterbiサーチ] |
| $t = 0, 1, .., T-1$においてstep2,step6を実行<br>2) $w_1 = 0, ..., Q-1$においてstep3を実行<br>3) $w_0 = 0, ..., Q-1$においてstep4を実行<br>4) $i = 0, 1, ..., l_{w_0} - 2$においてstep5を実行<br>5) $G_t(w_1, w_0, i) =$<br>$\max(G_{t-1}(w_1, w_0, i) \times a_{i,i}^{w_0} \times b_i^{w_0}(O_t),$<br>$G_{t-1}(w_1, w_0, i-1) \times a_{i-1,i}^{w_0} \times b_{i-1}^{w_0}(O_t))$ |
| [単語境界の計算] |
| 6) $w_1 = 0, 1, ..., Q-1$においてstep7を実行<br>7) $w_0 = 0, 1, ..., Q-1$においてstep8を実行<br>8) $\Delta = \max_{0 \le w_2 \le Q-1}(G_{t-1}(w_2, w_1, l_{w_1} - 2)$<br>$\times a_{l_{w_1}-2, l_{w_1}-1}^{w_1} \times b_{l_{w_1}-1}^{w_1}(O_t) \times P(w_0\|w_2, w_1)^\alpha$<br>もし$\Delta \ge G_t(w_1, w_0, 0)$ならば$G_t(w_1, w_0, 0) = \Delta$ |

ところで、従来から文音声認識のためのアルゴリズムとして2段DPや、one-pass DP、level building などのアルゴリズムが提案されている。このうち one-pass DP は各認識単語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音響モデルと言語の trigram モデルが簡単に結合できる。ただし、trigram は2つ前の単語が決定されて初めて現在の単語の出現確率が計算できるため、one-pass DP の内部状態においては、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1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 2Word で、単語の HMM は 4-state 3-loop で、状態は1から3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。

①は時間 $t-1$ において現在の語が $w_2$ で前の語が $w_2$ で状態が 1、②は時間 $t-1$ において現在の語が $w_2$ で前の語が $w_2$ で状態が 2、③は時間 $t$ において現在の語が $w_2$ で前の語が $w_2$ で状態が 2、④は時間 $t-1$ において現在の語が $w_2$ で前の語が $w_2$ で状態が 3、⑤は時間 $t-1$ において現在の語が $w_2$ で前の語が $w_1$ で状態が 3、⑥は時間 $t-1$ において現在の語が $w_1$ で前の語が $w_2$ で状態が 1、⑦は時間 $t$ において現在の語が $w_1$ で前の語が $w_2$ で状態が 1 までの最大累積尤度であるとする。

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。

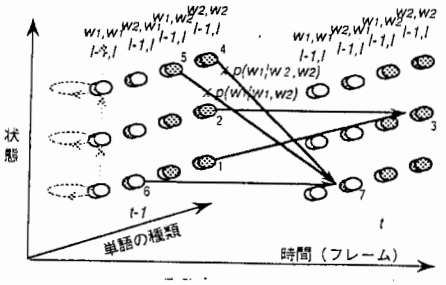


図 1: 単語の trigram を用いた Viterbi サーチのアルゴリズム

## 3 計算量およびメモリ量を減少させた文音声認識アルゴリズム

表1に示したアルゴリズムを実行する場合、大量のメモリが必要である。そこでメモリ量と計算量を減らすために、次に述べるようなアルゴリズムの改良をした。これらの改良により、HP730 を使用して認識単語数が 1500 のとき、メモリ量 15Mbyte 平均文認識時間平均 1 分 30 秒で実行可能になった。

### 3.1 ビームサーチ

one-pass DP は、全ての尤度の計算が終了した後に、累積尤度が最大の文候補を出力する。したがって各フレームにおいて累積尤度の低い文候補は、以後の探索から除外できる可能性が高い。つまり、フレームごとに、最尤なものから、ある個数（ビーム幅 $b$）のみを計算することにより、多くの場合最適性が保証される。このビームサーチを行なうことにより、計算量およびメモリ量が大幅に減らせる [12]。具体的には、すべての $w_1, w_0, i$ に対して step5 の式の計算を行なわずに、高い最大累積尤度を持つものから、ある個数（ビーム幅 $b$）のみを計算する。

このビームサーチを利用すると、フルサーチが、$G_t(w_1, w_0, i)$ を記憶するために、[認識語彙数 $^2$× 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 $b$] しか必要としなくなるため、必要なメモリ量が大幅に減少する。同様な比率で計算量の削減も可能になる。ただし、ビーム幅などが大きい場合、ビーム幅の選択方法の計算量の増加などの理由により、ビームサーチを行なわない方が計算が早く

なる場合がある。

### 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. 尤度のしきい値をもって、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が、認識を行なう前に閾値を決めておく必要がある。また、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るなどの現象がおきる可能性がある。一定の個数を残す方法は、フレームごとのソーティングが必要になるため、これが大きな計算量の負荷になると考えられてきた。しかし、ビームサーチでは上位からビーム幅の個数を示す尤度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 $O(N\log_2 N)$ の計算が必要であるが、ここでは、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。ここでは、後者を選択した。

### 3.3 Viterbi サーチの経路計算

Viterbi サーチでは、最尤度のワード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。そして認識終了後トレースバックを行なう [4]。この方法は、各時刻、各状態において、選択した経路を残すために［最大認識単語数$^2$×最大の単語の状態数×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、全ての尤度の計算が終了した後に最大累積尤度のワード列の結果をトレースバックするため、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、最大累積尤度を計算したときに、同時に、選択した経路を次の状態に渡す [8]。

   この場合、［最大認識単語数$^2$×最大の単語の状態数×文を構成する最大単語数］のメモリ量が必要である。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、必要なメモリ量は削減できる。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。ここでは、メモリ量の増加を押えるため、後者を選択した。

### 3.4 認識単位・音素

表1に示したアルゴリズムは、基本的には連続単語認識アルゴリズムである。ここで音素の HMM が連続的に接続されて、単語の HMM は構成されていると考える。例えば「通訳」という単語の HMM は /ts/,/u/,/y/,/a/,/k/,/u/ の計6音素の HMM が連結されて構成されているとする。

そして、認識単位を単語として各単語のHMMのシンボル出力確率を計算するかわりに、認識単位を音素として各音素のHMMのシンボル出力確率を計算し、単語のシンボル出力確率はこの値をコピーすることによって、同様な結果が得られる。これにより計算量が削減できる。

### 3.5 trigram の値の記憶

trigram の値を直接記憶する場合［最大認識単語数$^3$］のメモリ量が必要である。しかし、サンプリングデータ中に存在する組み合わせをリスト構造で記憶することにより、メモリ量を削減できる。また、完全ハッシュアルゴリズムを採用することにより、trigram の値を参照するための計算量は少なくてすむ。

## 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 をもち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2に示す。なお、音

声データの前後には約20msのポーズが付加されている。実験文数は262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2: 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 | 3-state4-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 | テストデータと同一話者の2670単語発声（特定話者認識）<br>男性話者12名の736単語発声（不特定話者認識） |
| 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 | 262文 |
| 発声様式 | 朗読発話 |
| 発声内容 | 国際会議の申し込み |
| 発声内容 | （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<br>171978単語 |

## 4.2 実験結果

実験結果を表3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78.6%、不特定話者認識では59.5%が得られた。またtrigramはbigramと比較すると高い認識性能が得られた。

表3: 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 | 63.4% (166/262) | 43.9% (115/262) |
| trigram | 78.6% (207/262) | 59.5% (156/262) |

text-closed ビーム幅:4096 $\alpha$:32

表4に、特定話者認識においてtrigramを利用したとき誤認識された文を示す。これからわかるように、助詞の誤りが多く、特に「ん」と「の」の誤りが目立つ。また、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90%であった。なお不特定話者認識の場合は約70%であった。

表4: 実験において誤りが出力された文

音声 open・テキスト closed 学習データ 171978 単語 ビーム 4096 $\alpha$:32

正解文 → 1位出力

助詞の誤り（意味的にはほぼ同じ） 16文

会議の宿泊施設についてお尋ねしたいのですが
→ 会議の宿泊施設についてお尋ねしたいんですが
それでは京都プリンスホテルを予約したいのですが
→ それでは京都プリンスホテルを予約したいんですが
電話番号もお願いします
→ 電話番号お願いします

類似した単語の誤り（意味的にはほぼ同じ） 9文

京都プリンスホテルが会議場には近いんですが
→ 京都プリンスホテルが会場には近いんですが
どのようなご用件でしょうか
→ どのような用件でしょうか
登録費としてお一人三万五千円が必要です
→ 登録費としてお一人3万5千円かかります

意味的に大きくことなった誤り 29文

住所は東京都港区新橋1丁目1番3号です
→ まず発表者の方コンピューター関係の雑誌の方が
してこの研究を発表してくださるのです
電話番号は３３１の２５２１です
→ 論文の発表は午前中の9時に会場に近いです
そちらでどこか紹介していただけますか
→ そちらで取っているとお話をお書きしていただけますか
私共でご紹介できるホテルは京都ホテルと京都プリンスホテルです
→ 私共の方で研究発表の登録料をお送りするのです
ではお名前とご住所をお願いします
→ ご発表になる方のご住所をお願いします

これらの誤出力された音声データのなかで、「住所は東京都港区新橋1丁目1番3号です」は「住所は」の後にポーズが145msあった。この音声区間では、音響モデルとしてはポーズであるが、言語モデルでは、このポーズを想定していないため、誤認識が起きたものと予想される。

## 4.3 ポーズの処理1 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、言語モデルではカバーしきれないため、ポーズの区間で誤認識が起きやすい。そこで全ての単語と単語の境界にポーズが入力されても、文

認識が可能な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。

アルゴリズムで示すと、表1 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2と同一である。この結果を表6に載せる。このポーズのスキップにより、特定話者認識では、認識性能が向上した(78.3% → 86.3%)。しかし、不特定話者認識では、認識性能はあまり向上しなかった(59.5% → 60.3%)。

表6: 認識実験の結果（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 | 68.3% (179/262) | 42.7% (112/262) |
| trigram | 86.3% (226/262) | 60.3% (158/262) |

text-closed ビーム幅:4096 $\alpha$:32

## 4.4 ポーズの処理2 ポーズの学習

不特定話者認識の実験では、音声データの最初のポーズ区間が認識できていないために誤認識が起きていることが多かった。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不特定話者認識の認識性能が向上した( 60.3% → 84.0%)。そして不特定話者認識の認識率は特定話者認識とほぼ同等な値が得られた。またtrigramの特定話者認識の実験において、意味的に正しいものを正解に含めたときの認識率は93.5%に達した。意味的に間違っている誤りは17文で、これを表8に載せる。

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 | 70.2% (184/262) | 57.6% (151/262) |
| trigram | 86.3% (226/262) | 84.0% (220/262) |

text-closed ビーム幅:4096 $\alpha$:32

表8: 実験において誤りが出力された文 音声 open・テキスト closed 学習データ 171978単語 ビーム 4096 $\alpha$:32

正解文 → 1位出力
私共でご紹介できるホテルは京都ホテルと京都プリンスホテルです
→ 私共の方からこのホテルが3つ京都ホテルと京都プリンスホテルです
電話番号は３３１の２５２１です
→ 論文の発表は午前中の９時に会場に近いです
京都プリンスホテルに８月４日から８日まで一人部屋をお取りしました
→ 京都プリンスホテルに８月４日から８日まで一人部屋をお送りしました
失礼します
→ そうします
それでは登録用紙をお送りいたします
→ それでは登録用紙をお受けいたします
今回は割引を行なっておりません
→ 今回は割引を行なっております
会議は８月２２日から２５日まで京都国際会議場で開催されます
→ 会議が８月５日の１２時までのその国際会議場の方で開催されます
会議の案内書をお送りいたしますのでそれをご覧ください
→ 会議の案内書をお受けいたしますのでそれをご覧ください
ベル研のジム・ワイベルです
→ 論文の図表を直したいのです
すでに登録料の８万５千円を振り込まれておられますね
→ すでに登録料の８万５千円を振り込まれておられます
では誰かが私の代わりに参加することはできますか
→ 連絡取りますが私の会議に参加することできますか
代理人が参加する場合はあらかじめこちらまでお知らせください
→ 会議に参加する場合はあらかじめこちらまでお知らせください
代理人が決まりましたらお知らせいたします
→ 時間がかかりましたらお知らせいたします
では失礼します
→ これは失礼いたします
いいですよ
→ いいです
それでは登録用紙をお送りいたします
→ それでは登録用紙をお受けいたします
どうも失礼いたします
→ どうもお手数おかけいたします

## 4.5 ビーム幅

ビームサーチは、最終的な累積尤度最大の文候補は、各フレーム時においても最大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2と同一である。また、4.3章および4.4章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および表10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、bigram、trigram、特定話者認識、不特定話者認識、いずれの場合もあまり認識性能は向上しないことがわかる。

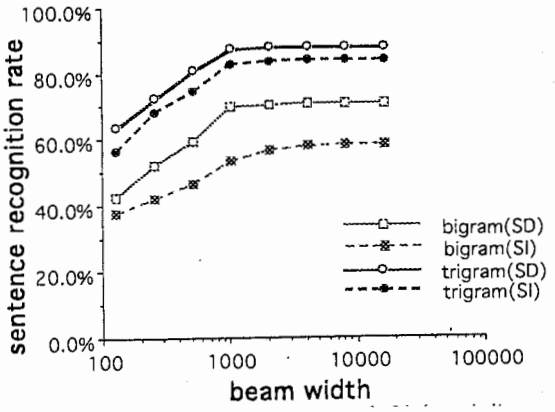


図2: ビーム幅を変化させたときの変化　認識率(%)
text-closed $\alpha$:32

## 4.6 音響尤度と言語の連鎖確率の結合値

言語情報として単語のtrigramを用いた場合、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$w_1, w_2, ..., w_N$を選択することである。

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

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$\alpha$を32としている。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2と同一である。また、4.3章および4.4章で述べたポーズ処理はおこなっている。

この結果を図3および表11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$\alpha$が32のとき最も高い文認識率を示した。

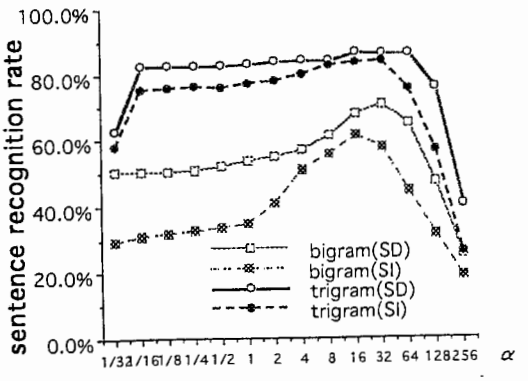


図3: 音響尤度と言語の連鎖確率の結合値を変えたときの認識性能の変化　認識率(%)
text-closed ビーム幅:4096

## 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るために、認識実験を行なった。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（ＡＴＲ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種類の実験を行なった。実験条件は、表2と同一である。また4.3章および4.4章で述べたポーズ処理はおこなった。

この実験結果を図4および表12および表13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。

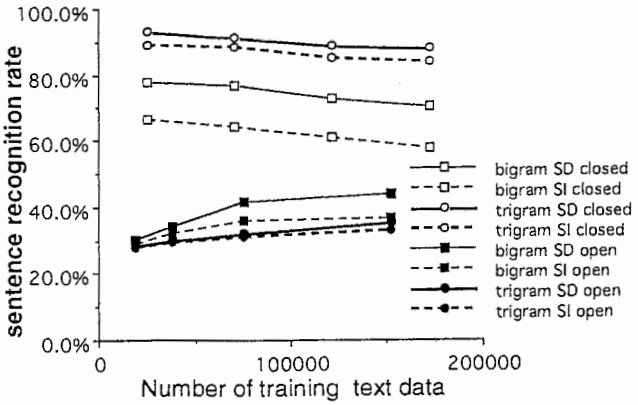


図 4: 学習データ量における認識結果の変化　認識率 (%)

## 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 [9]。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。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。

### 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に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、個人的には、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ている。ここでは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、話者はナレータではなくて一般の人である。

### 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、ポーズの処理は行なっている。その他の実験条件は、表 2 と同様である。この実験結果を表 9 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 9: 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　学習データ

今回の実験で用いた trigram の連鎖確率値の計算には、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 万単語を利用した。このデータベースにおいて、学習量に対するマルコフ連鎖確率値の変化をみるために、

エントロピーと“頻度別出現率”も調査した。“頻度別出現率”とは次のように定義する。

“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。これから、エントロピーは、また安定な値に収束していないことがわかる。このデータベースでは全体の語彙の 58.8%(3486/5933) は 1 回しか出現していない。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い。単語 bigram の全ての組合せの中の 36.5%(13674/37752) は 1 回しか出現していない。これらの結果はデータ量の不足を示している。つまり、実験に使用した trigram の連鎖確率値は、信頼のある値であるとは言えない。

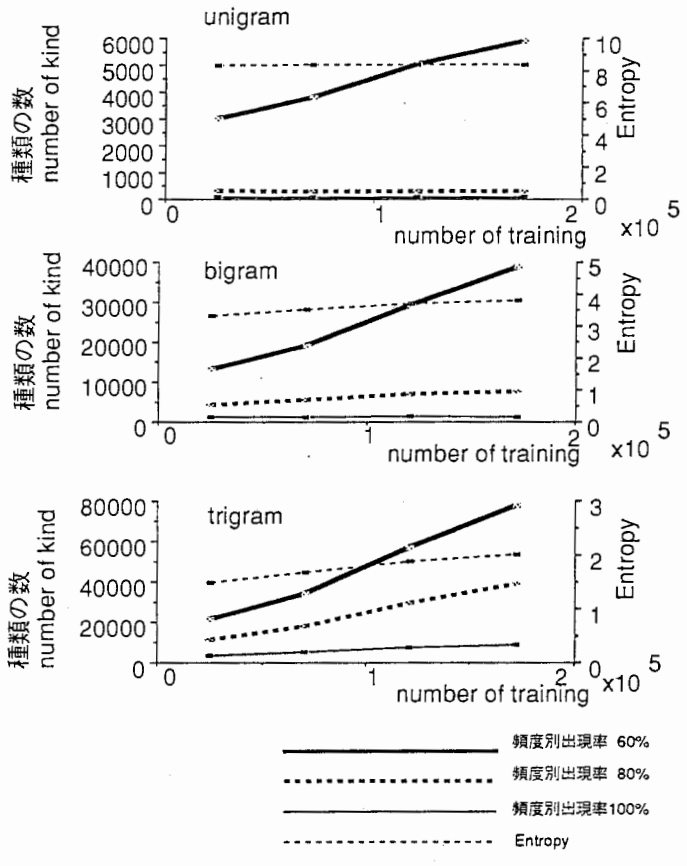


図 5: 学習データの入力データに対するエントロピーおよび頻度別出現率の変化

## 6.2 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。実験ではビーム幅が 1024 を越えると、bigram、trigram、特定話者認識、不特定話者認識、いずれの場合もあまり認識性能は向上しないことがわかった。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かもしれない。ただし、ここで実験に用いた話者はナレータであるため、非常にクリーンな音声データと言える。したがって通常の話者では、このビーム幅では不足する可能性がある。

## 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 $\alpha$ が 32 のとき最も高い文認識性能が得られることがわかった。

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 $\alpha$ 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 $\alpha$ は 1 にすべきかもしれない。

## 6.4 text-open data の認識率

この実験では text-open data と text-closed data において認識性能に大きな差が生じた。また、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かった。この原因として、次の理由が考えられる。言語モデルとして bigram や trigram などのような統計モデルを用いた場合、基本的に、text-open data の認識性能は、学習データとテストデータの相対的なカバー率に依存するため、テストデータの選択方法によって認識率は大きく変化する。そのため text-open data の認識率の信頼性に関しては十分に注意する必要があり、大量の text-open data を処理しないかぎり認識率の信頼性は低いと考えられる。音素認識などで行なわれているように、大量のテキストと音声データを奇数と偶数に分離し、各々を学習データとテストデータにするような実験が必要であるが、この実験を行なうには大量のテキストと音声データが必要である。

### 6.5 言語モデルについて

ここで提案したアルゴリズムは、計算量は、ビーム幅に大きく依存し、言語モデルには、あまり依存しない。従って、text-closed data の認識に限れば、言語モデルの Markov モデルの次数を上げることにより、認識性能は向上できる。従って、今後の言語モデルの研究として、任意のテキストにおいて、高いカバー率と低い perplexity を持った言語モデルの研究が必要であろう。

### 6.6 自由発話認識と text-open

自由発話の認識実験から、単語 trigram を利用して、冗長語が伴う text-closed data の音声を認識した場合、つまり trigram の連鎖確率値の計算に使用したテキストデータを認識する場合、高い認識性能が得られた。したがって自由発話でも text-closed data ならば高い認識性能が得られることが予想される。しかし、一般的に自由発話は text-open data になる。仮にテキストデータが大量にあれば、text-open data と text-closed data の認識率は接近すると思われるが、今回の実験ではテキストデータは小規模である。したがって、今回計算した trigram の連鎖確率値の一般性は低い。そのため text-open data の認識性能は低い。

また、text-open data では未知語が出現する。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 one-pass DP 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。朗読発話の文認識実験の結果、不特定話者認識においては、text-closed data では 59.5% 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、言語モデルではカバーしきれないため、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。ポーズに対してこれらの対策をすることにより、84.0% 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。

## 参考文献


[1] 中川 聖一, 大黒 慶久, "連続音声認識における音韻認識率と文認識率との関係", 電子情報通信学会論文誌, D-2,Vol.J72-D-2,No.2 PP.207-217 (1989).

[2] Kenji KITA, "HMM CONTINUOUS SPEECH RECGNITION USING PREDICTIVE LR PARSING," Vol.2, S13.3 PP.703-706 ICASSP (1989).

[3] A.Averbuch, "An IBM PC BASED LARGE-VOCABULARY ISOLATED-UTTERANCE SPEECH RECOGNIZER," Vol.1, 2.4.1 PP.53-56 ICASSP (1986).

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

[5] Kiyoshi Shikano, "IMPROVEMENT OF WORD RECOGNITION RESULTS BY TRIGRAM MODEL," Vol.3, 29.2.1 PP.1261-1262 ICASSP(1987).

[6] Tomohiro Yamada, Shouichi Matsunaga, Kiyohiro Shikano, "JAPANESE DICTATION SYSTEM USING CHARACTER SOURE MODELING," Vol.1, 1-37 ICASSP(1992).

[7] Isao Murase , Seiichi Nakagawa "Sentence Recognition Method Using Word Cooccurrence Probability and Its Evaluation," Vol.2, pp. 1217-1220 ICSLP90 (1990).

[8] 村上 仁一、他: "2重マルコフ連鎖確率モデルを使用した単音節音声入力の改善," 信学技報, SP88-29, pp.63-70 (June 1988).

[9] 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).

[10] 鹿野 清宏, "Trigram Model による単語音声認識結果の改善," , 電子情報通信学会技術報告,SP87-23, pp.9-16 (1987).

[11] 南 泰浩、中川 正雄, "trigram モデルを用いた複数候補を求めるフレーム同期型 HMM 連続音声認識," 電子情報通信学会論文誌, D-2,Vol.j73-D-2,No9 PP.1383-1392 (1990).

[12] 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)

[13] 小林、他: "間投詞、言い直し等の出現に関する音響的特徴," SPREC-93-1, pp.7-10 (July 1993).

[14] 高木、他: "対話における話題展開と発話単位の性質," SPREC-93-1, pp.11-18 (July 1993).

付録

表 10: ビーム幅を変化させたときの変化　認識率 (%)

特定話者認識:SD 不特定話者認識:SI

| beam width | SD (bigram) | SI (bigram) | SD (trigram) | SI (trigram) |
|---|---|---|---|---|
| 128 | 42.3% (111/262) | 37.4% ( 98/262) | 63.7% (167/262) | 56.5% (148/262) |
| 256 | 51.9% (136/262) | 41.9% (110/262) | 72.5% (190/262) | 68.3% (179/262) |
| 512 | 59.1% (155/262) | 46.1% (121/262) | 80.1% (212/262) | 74.4% (195/262) |
| 1024 | 69.8% (183/262) | 53.0% (139/262) | 87.4% (229/262) | 82.8% (217/262) |
| 2048 | 70.2% (184/262) | 56.4% (148/262) | 87.8% (230/262) | 83.6% (219/262) |
| 4096 | 70.6% (185/262) | 57.6% (151/262) | 87.8% (230/262) | 84.0% (220/262) |
| 8192 | 70.6% (185/262) | 58.0% (152/262) | 87.8% (230/262) | 84.0% (220/262) |
| 16384 | 70.6% (185/262) | 58.0% (152/262) | 87.8% (230/262) | 84.0% (220/262) |

trigram text-closed $\alpha$: 32

表 11: 音響尤度と言語の連鎖確率を変えたときの認識率の変化　認識率 (%)

特定話者認識:SD 不特定話者認識:SI

| $\alpha$ | SD (bigram) | SI (bigram) | SD (trigram) | SI (trigram) |
|---|---|---|---|---|
| 1/32 | 50.3% (132/262) | 29.3% ( 77/262) | 76.0% (165/217) | 58.0% (152/262) |
| 1/16 | 50.3% (132/262) | 30.9% ( 81/262) | 82.8% (217/262) | 75.6% (198/262) |
| 1/8 | 50.3% (132/262) | 31.6% ( 83/262) | 82.8% (217/262) | 76.0% (199/262) |
| 1/4 | 50.7% (133/262) | 32.8% ( 86/262) | 82.8% (217/262) | 76.3% (200/262) |
| 1/2 | 51.9% (136/262) | 33.5% ( 88/262) | 82.8% (217/262) | 76.0% (199/262) |
| 1 | 53.4% (140/262) | 34.7% ( 91/262) | 83.2% (218/262) | 77.1% (202/262) |
| 2 | 54.9% (144/262) | 40.8% (107/262) | 84.0% (220/262) | 77.9% (204/262) |
| 4 | 56.8% (149/262) | 50.7% (133/262) | 84.4% (221/262) | 80.0% (209/262) |
| 8 | 61.4% (161/262) | 55.7% (146/262) | 84.4% (221/262) | 82.8% (217/262) |
| 16 | 67.5% (177/262) | 61.4% (161/262) | 86.6% (227/262) | 83.6% (219/262) |
| 32 | 70.6% (185/262) | 57.6% (151/262) | 86.6% (226/262) | 84.0% (220/262) |
| 64 | 64.8% (170/262) | 44.2% (116/262) | 86.6% (226/262) | 75.6% (198/262) |
| 128 | 47.3% (124/262) | 31.2% ( 82/262) | 80.0% (199/262) | 56.8% (149/262) |
| 256 | 25.5% ( 67/262) | 19.0% ( 50/262) | 40.5% (106/262) | 26.3% ( 69/262) |

trigram text-closed beam 幅:4096

表 12: 学習データ量における認識結果の変化 text-closed　認識率 (%)

特定話者認識:SD 不特定話者認識:SI

| 学習データ数 (単語数) | SD (bigram) | SI (bigram) | SD (trigram) | SI (trigram) |
|---|---|---|---|---|
| 25071 | 77.8% (204/262) | 66.4% (174/262) | 93.1% (244/262) | 89.3% (234/262) |
| 70336 | 76.7% (201/262) | 64.1% (168/262) | 91.2% (239/262) | 88.2% (231/262) |
| 120661 | 72.5% (190/262) | 60.6% (159/262) | 88.5% (232/262) | 85.1% (223/262) |
| 171978 | 70.2% (184/262) | 57.6% (151/262) | 87.8% (230/262) | 84.0% (220/262) |

trigram text-closed beam 幅:4096 $\alpha$: 32

表 13: 学習データ量における認識結果の変化 text-open　認識率 (%)

特定話者認識:SD 不特定話者認識:SI

| 学習データ数 (単語数) | SD (bigram) | SI (bigram) | SD (trigram) | SI (trigram) |
|---|---|---|---|---|
| 19092 | 30.1% ( 79/262) | 29.0% (76/262) | 28.6% (75/262) | 27.8% (73/262) |
| 37847 | 34.3% ( 90/262) | 32.4% (85/262) | 30.1% (79/262) | 29.3% (77/262) |
| 75487 | 41.6% (109/262) | 35.8% (94/262) | 32.4% (85/262) | 30.9% (81/262) |
| 151534 | 43.8% (115/262) | 36.6% (96/262) | 35.4% (93/262) | 33.2% (87/262) |

trigram text-open beam 幅:4096 $\alpha$: 32