TR-IT-0099

単語の $N-gram$ を利用した音声認識アルゴリズムと自由発話認識

A Spontaneous Speech Recognition Algorithm Using Word Trigram Model

村上仁一
Jin'ichi Murakami

1995.2

概要

本論文では単語の n-gram モデルを使用した連続音声認識システムの概要と自由発話認識の問題点と解決方法について述べる。自由発話を認識するにあたって、特に問題になるのは、冗長語（間投詞）や言い淀み、言い直しである。このような現象は、認識性能が高い音響モデルを作成することを困難にすると考えた。そこで本論文では特に言語モデルに着目した。そして仮名や漢字や単語の N-gram を利用することを考えた。言語の N-gram モデルは確率モデルの中で最も基本的なモデルである。しかし、確率付き文脈自由文法などの他の言語モデルと比較すると、Perplexity が最も低いと考えている。この $N-gram$ を用いて自由発話認識を行なった。この実験結果について述べる。

©ATR 音声翻訳通信研究所
©ATR Interpreting Telecommunications Research Labs.

単語の $N-gram$ を利用した音声認識アルゴリズムと自由発話認識

村上仁一
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

1995/3/1

目次

第 1 章

序論

本論文では単語の n-gram モデルを使用した連続音声認識システムの概要と自由発話認識の問題点と解決方法について述べる。

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」などに代表される間投詞や、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。

一方、音声認識システムにおいては、音響処理だけでは認識性能が低いため、言語処理が利用されている。そして言語モデルと Perplexity と認識性能には経験的に相関があり、Perplexity が低い言語モデルほど認識性能が高いことが知られている。音声認識のための言語モデルには多くの言語モデルがあるが、大きく分類してルールベースの言語モデルと確率ベースの言語モデルがある。

ルールベースの言語モデルとして、ネットワーク文法や文脈自由文法、Unification 文法などがあげられる。これらの言語モデルの問題点として、人間がルールを記述するため、文法を書く負荷が大きいことがあげられる。そのため、文法をメインテナンスをすることは困難である。また、詳細なルールを書くことが困難であるため、これらの言語モデルでは非文を生成しやすい傾向がある。

確率ベースの言語モデルとして単語の N-gram や確率付きネットワーク文法や確率付き文脈自由文法などがあげられる。確率付きネットワーク文法とはネットワーク文法に確率を付けたモデルで、大量に解析されたテキストデータがあればこれから確率を直接計算できる。また、Ergodic HMM を想定した場合、Baum-Welch アルゴリズムを使用することによって大量のテキストデータベースから確率が直接計算できる [34]。同様に確率付き文脈自由文法とは文脈自由文法に確率をつけたモデルであるが、これも Inside-Outside アルゴリズムを利用することで大量のテキストデータベースから値を求めることができる [43]。

自由発話を認識するにあたって、特に問題になるのは、冗長語（間投詞）や言い淀み、言い直しである。このような現象は、認識性能が高い音響モデルを作成することを困難にすると考えた。そこで本論文では特に言語モデルに着目した。そして仮名や漢字や単語の N-gram を利用することを考えた。言語の N-gram モデルは確率モデルの中で最も基本的なモデルである。しかし、確率付き文脈自由文法などの他の言語モデルと比較すると、Perplexity が最も低いと考えている。

本論文では次のフローチャートに従ってのべる。

1. 言語のマルコフモデルにおけるエントロピーと収束性

2. 自由発話の音響的、言語的特徴

3. 連続音声認識のアルゴリズム

4. シュミレーションによる音節や漢字の trigram の有効性

5. 単語の HMM と単語の bigram を用いた文節認識アルゴリズム

6. 単語の trigram を用いた文認識アルゴリズム

7. 自由発話認識のためのアルゴリズムとその実験結果

第 2 章

学習データ量とマルコフ連鎖確率値の収束性について（trigram の分布）

2.1 エンロトピーと頻度別出現率

学習データ量の変化に対するマルコフ連鎖確率の値の変化を調べるために、まず学習データ量に対するエントロピーの収束率を調査した。unigram・bigram・trigram・4-gram のエントロピーは次の式によって計算できる。

$unigram \quad \Sigma_i p(w_i) log[p(w_i)]$
$bigram \quad \Sigma_{i,j} p(w_i, w_j) log[p(w_j|w_i)]$
$trigram \quad \Sigma_{i,j,k} p(w_i, w_j, w_k) log[p(w_l|w_i, w_j)]$
$4-gram \quad \Sigma_{i,j,k,l} p(w_i, w_j, w_k, w_l) log[p(w_l|w_i, w_j, w_k)]$

ここではエントロピーの他に“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 98%”が示す値は、学習データの中で 98% 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 96%、頻度別出現率 98%、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。

なお以後の図 2.1から 2.6 までの横軸は学習データ量で、縦軸は出現したマルコフ連鎖確率の種類の数およびエントロピーの値である。また図中における太い実線は頻度別出現率 96%、太い断線は頻度別出現率 98%、細い実線は頻度別出現率 100%、細い断線はエントロピーを示している。また“Entropy”の横に示した値は、全学習データを利用したときのエントロピーの値である。

2.2 新聞記事

標準的な日本語として 1982 年 1 月 4 日から 3 月 30 日までの 64 日分の日経新聞の新聞記事を選んだ。この記事を日本語形態素解析プログラムで形態素解析を行ない、音節と品詞を自動的に付与した。そして、文節に区切り、このデータから音節および漢字仮名および品詞のマルコフモデルの収束性を調べた。ただし、過度の複雑さを避けるため、記号・外国語読み・数詞の文字が存在する文は文全体を削除した。データ量は漢字仮名の文字数にして約 170 万文字である。また、使用した日本語形態素解析プログラムの形態素解析の精度は単語認定率で約 95% である [32]。表 2.1に新聞記事の一部を載せる。

表 2.1: 新聞記事の例

大蔵省はことし四月から新銀行法が施行されるのに伴い、在日外銀の営業活動を日本の銀行同様に扱うとの基本方針を決め、これを盛り込んだ政令を二月中にも公布する。おもな内容は（1）企業向け貸し出しに対する大口融資規制を在日外銀にも適用し、五年間の猶予期間を設けるなどの配慮をする（2）利益準備金の積み立てを義務づけ、外銀に対する信頼を高める（3）邦銀の支店を買収することや現地法人化を認める――など。大蔵省はこれによって在日外銀に関する法的根拠が明確になるほか、在日外銀の国内活動がしやすくなり、欧米諸国の間に出始めているわが国の金融制度に対する不満を和らげるのに役立つとみている。（在日外国銀行は「きょうのことば」参照）

2.2.1 新聞記事における音節のマルコフ連鎖確率の収束率

音節の unigram・bigram・trigram・4-gram の学習データ量に対するエントロピーおよび頻度別出現率のグラフを図 2.1に示す。音節の種類の数は、外来語を除き鼻音化したガ行を加え長音を 1 音節として 111 種類である。これらから以下のことがわかる。

1. エントロピーは比較的少ないデータで収束する。
2. 頻度別出現率 98% や 96% が収束するのに必要な学習データの量は、エントロピーを収束させるのに必要な学習データの量よりも多くのデータが必要である。
3. 頻度別出現率 100% は学習データを増やしても収束する傾向がみられない。これは、学習データを増加させるにともない、全体に占める割合は少ないが、新しいマルコフモデルの組み合わせがたえず出現することを意味している。
4. エントロピーは unigram・bigram・trigram・4-gram になるにしたがい低下する。

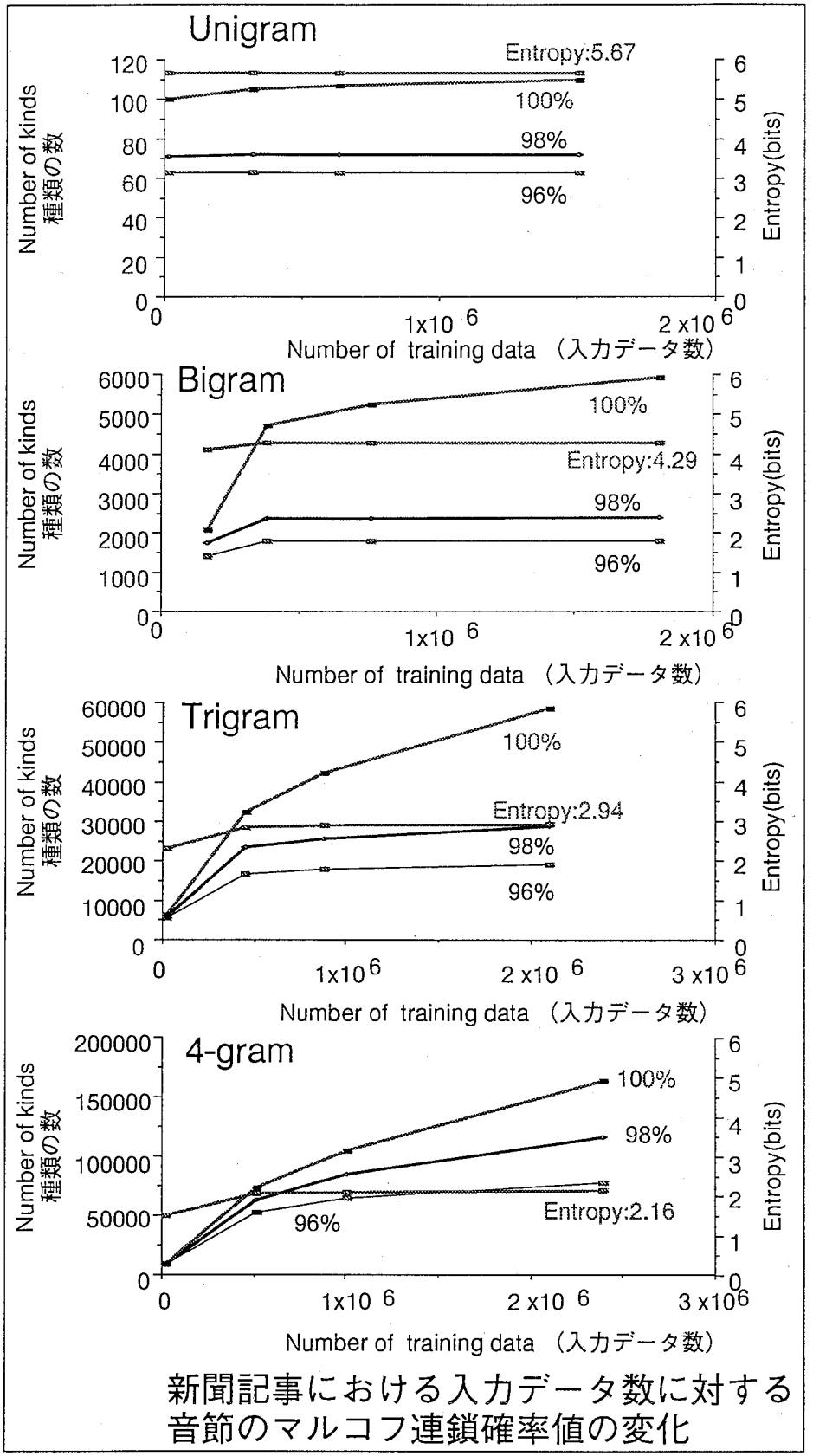


図 2.1: 新聞記事における学習データ数に対する音節のマルコフ連鎖確率値の収束率

2.2.2 新聞記事における漢字仮名文字のマルコフ連鎖確率の収束率

新聞記事における漢字仮名文字の学習文字数に対するエントロピーおよび頻度別出現率のグラフを図2.2に示す。なお、使用した漢字仮名の種類はJIS 1級、約3000種類に限定した。これらから以下のことがわかる。

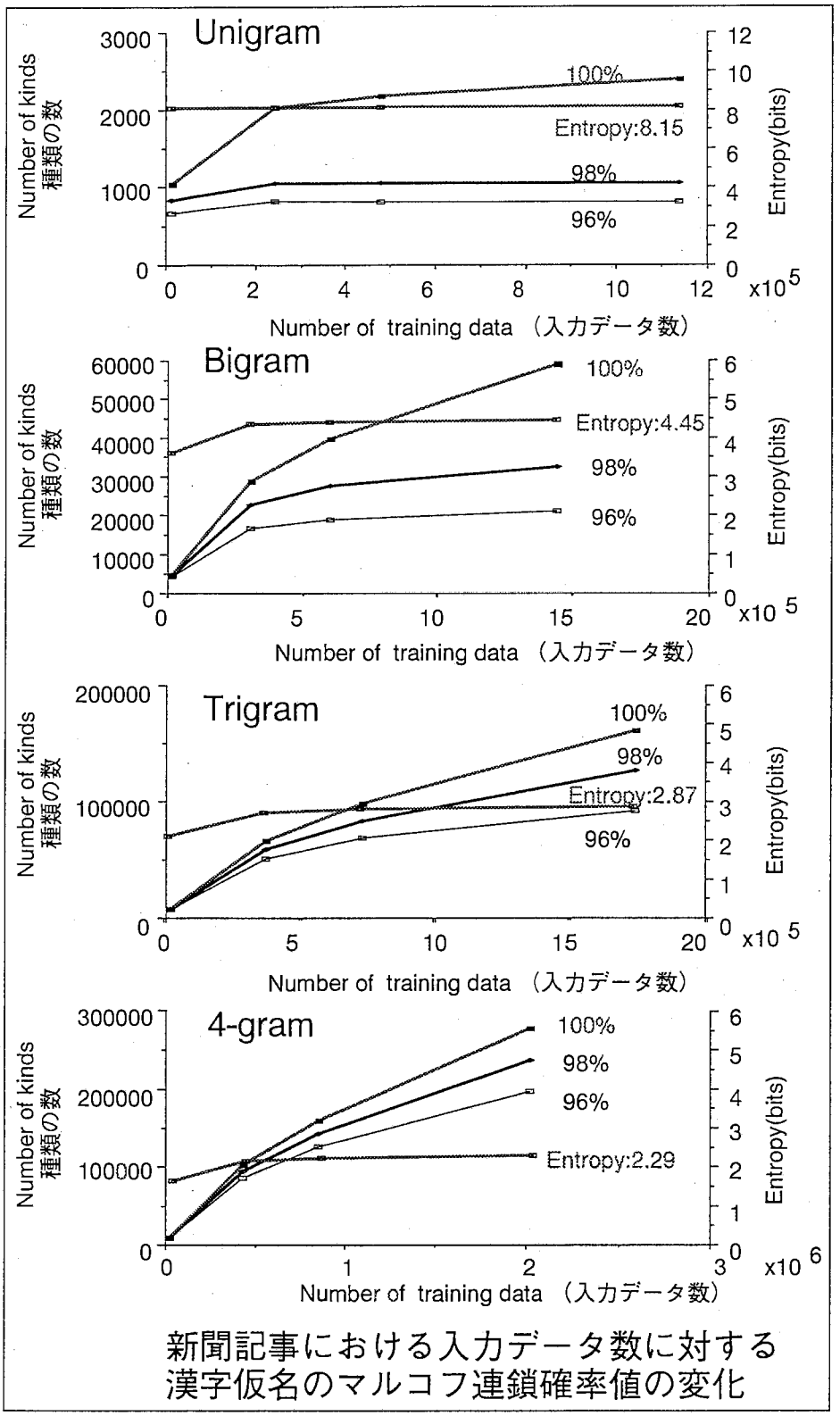


図 2.2: 新聞記事における学習データ数に対する漢字仮名のマルコフ連鎖確率値の収束率

1. 漢字仮名文字の場合、連鎖確率の値を収束させるためには音節の場合よりも大量のデータが必要である。

2. 頻度別出現率 98%,96% の収束に必要な学習データの量は、音節と同様にエントロピーの場合よりも多く必要である。

3. 漢字仮名と音節のエントロピーの値を比較すると、unigram と bigram においては、音節のエントロピーの方が低いが、trigram では漢字仮名文字のエントロピーの方が低い。漢字仮名の種類の数は音節の種類の数の約 30 倍もあることを考えると、漢字仮名文字の trigram の持つ情報量は、音節と比較すると、かなり多いと思われる。

2.2.3 新聞記事における品詞のマルコフ連鎖確率の収束率

品詞は、名詞・助詞などの機能的な分類の他に地名・人名・色の種類など意味的にも分類されていて、約 450 種類ある。学習データの量の変化に対する品詞のエントロピーおよび頻度別出現率のグラフを図 2.3に示す。これらから以下のことが示される。

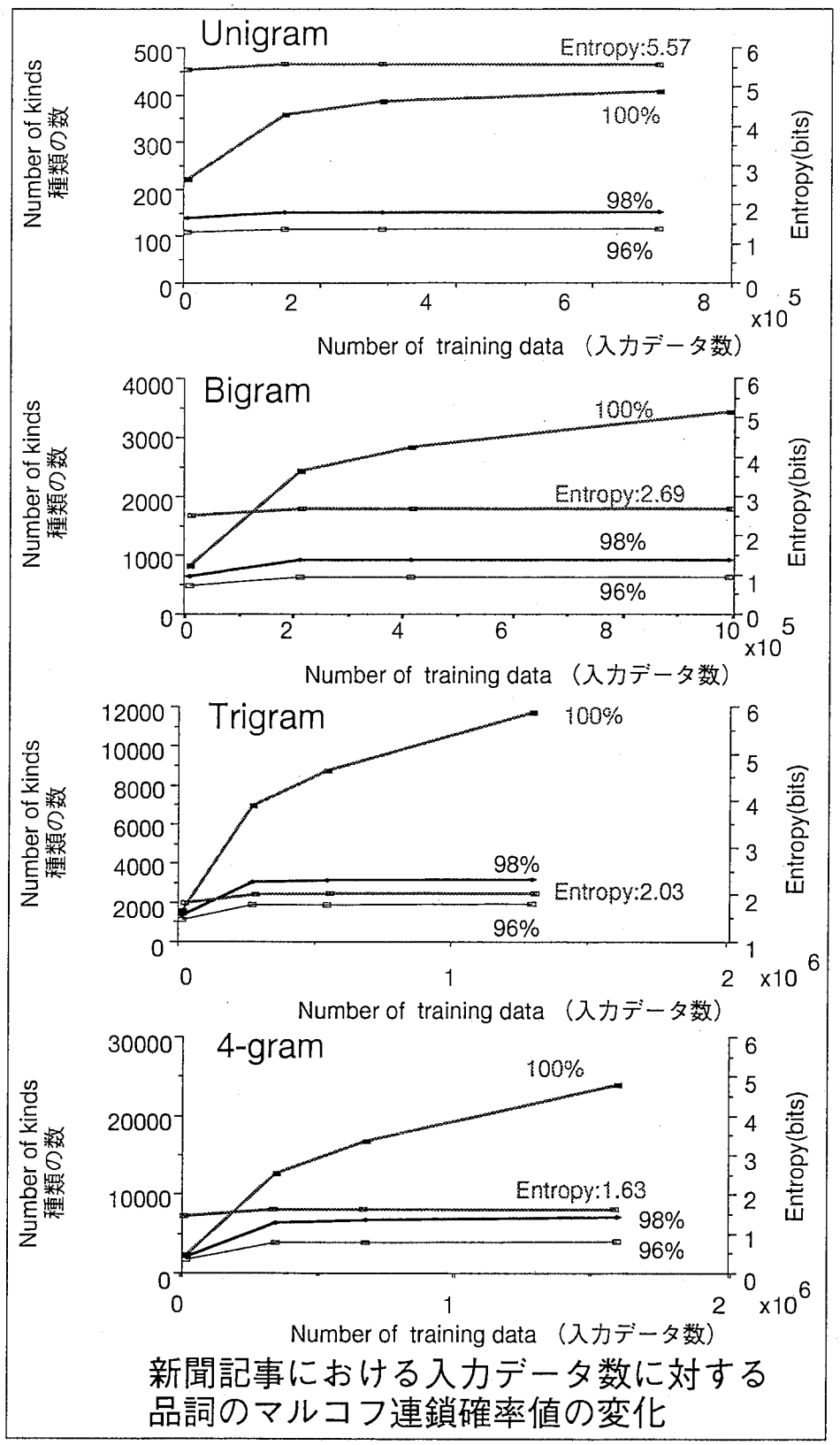


図 2.3: 新聞記事における学習データ数に対する品詞のマルコフ連鎖確率値の収束率

1. 品詞は、音節や漢字仮名と比較すると小量のデータで収束する。

2. 音節や漢字仮名では、unigram,bigram,trigram になるにしたがいエントロピーは半減している。しかし、品詞の場合、unigram のエントロピーの値に対して bigram のエントロピーの値は約半減するが、bigram のエントロピーの値に対して trigram のエントロピーの値は、あまり減少しない。したがって品詞の trigram の情報量は、少ないと思われる。

2.3 X 線 CT 所見作成

次に日本語の専門的な文章の例として X 線 CT 所見作成の文章を調べた。表 2.2に文の一部を載せる。

表 2.2: X 線 CT 所見作成の例

頭部ＣＴ 単純および造影
１、３月１３日のＣＴと比較した。
２、スライスのレベルが若干異なっているので正確な比較はできないが、鞍上槽の正中からやや右上方へ向かって進展している増強効果を示す腫瘤の大きさは本質的に変わっていない。ただし前回のＣＴでこの結節性腫瘤の右前方に見られた嚢胞性の成分については今回は描出されていない。３、側脳室の大きさ形も前回と同様である。

impression.....
鞍上槽の頭蓋咽頭腫の残存については明らかな変化はないが、右後方に見られた嚢胞性成分が消失しているかもしれない。

2.3.1 X 線 CT 所見作成における音節のマルコフ連鎖確率の収束率

X 線 CT 所見作成の文章は"mass effect","large magna", などの外来語が数多く出現する。そのため音節の種類の数は 118 種類になった。図 2.4に学習データ量に対する音節の unigram・bigram・trigram およびエントロピーの値の変化を示す。

2.3.2 X 線 CT 所見作成における漢字仮名のマルコフ連鎖確率の収束率

X 線 CT 所見作成の文章には外来語が多く出現する。ここでは、これらの外来語を全て 1 文字の全角文字として（例えば “mass effect” は MASS EFFECT）漢字仮名のマルコフ連鎖確率の収束性を調べた。この結果を図 2.5に示す。

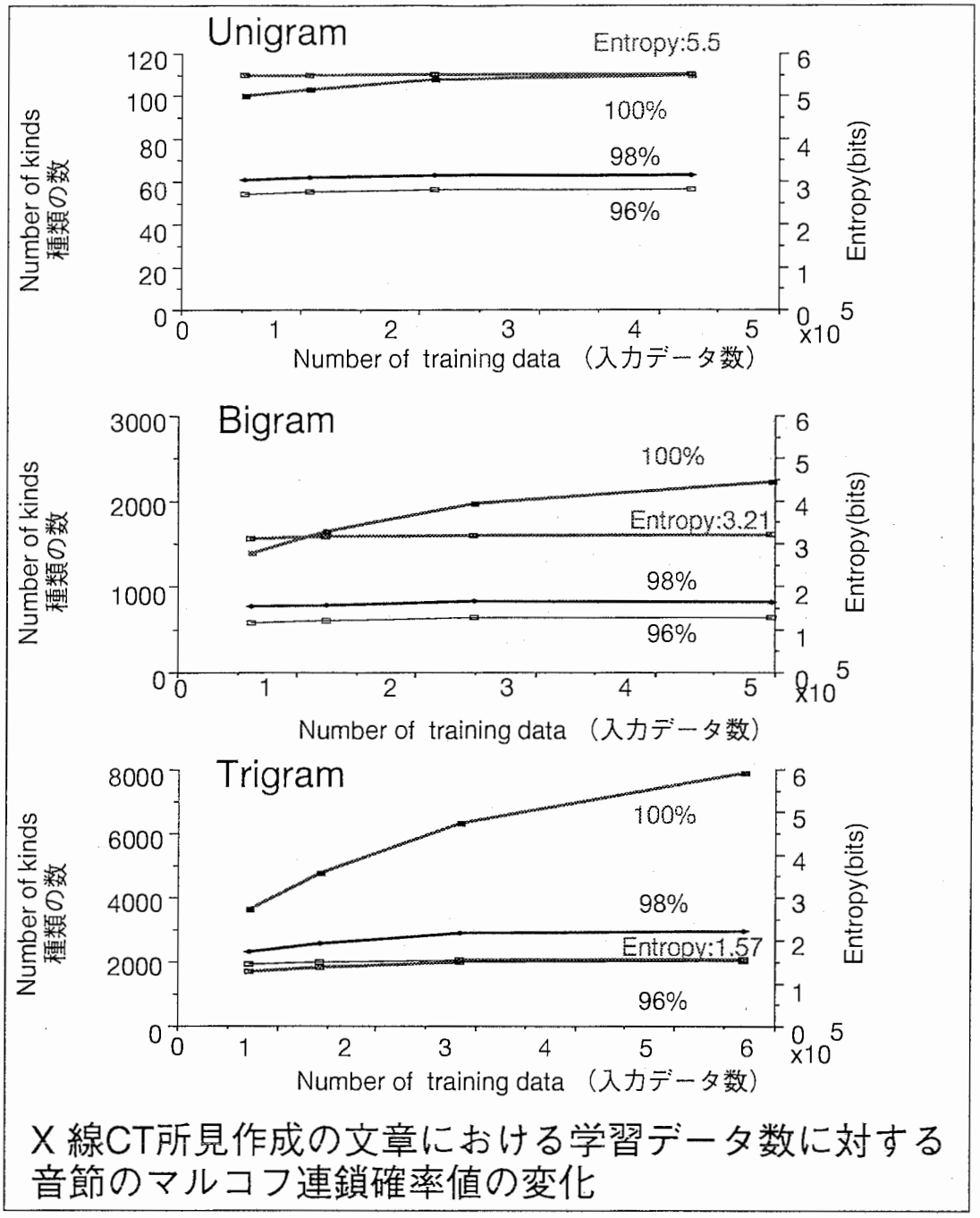


図 2.4: X 線 CT 所見における学習データ数に対する音節のマルコフ連鎖確率値の収束率

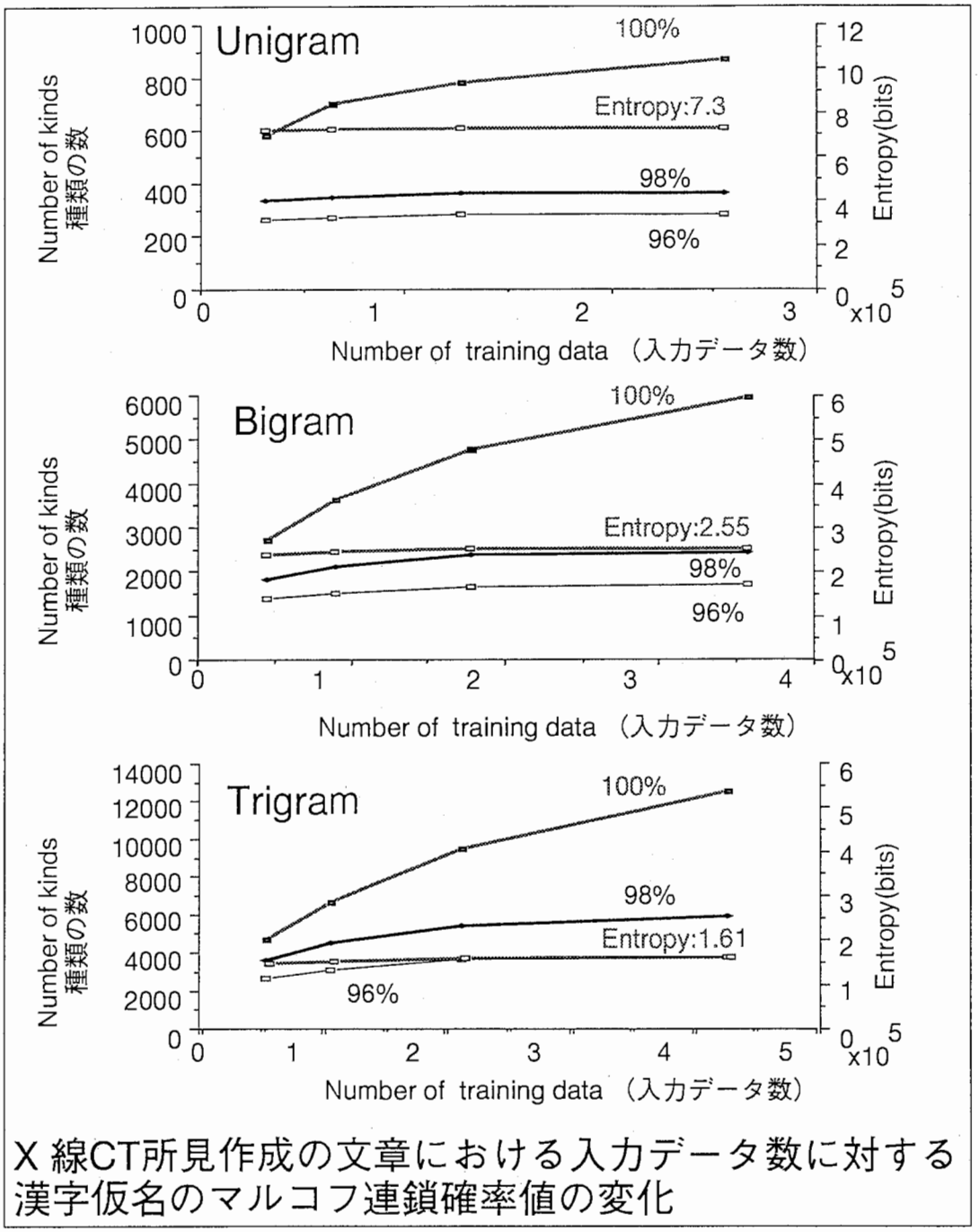


図 2.5: X 線 CT 所見における学習データ数に対する漢字仮名のマルコフ連鎖確率値の収束率

2.3.3 X 線 CT 所見作成における単語のマルコフ連鎖確率の収束率

X 線 CT 所見作成の文章の語彙数は約 3000 語である。ただし、全体の認識性能を向上させるため文節出現率が高いものから上位 100 文節は単語として登録してあるため、通常、文節と考えられるものまで単語と見なしている（例えば”脳実質を”は 1 単語）。X 線 CT 所見作成における単語のマルコフ連鎖確率の収束性を図 2.6 に示す。

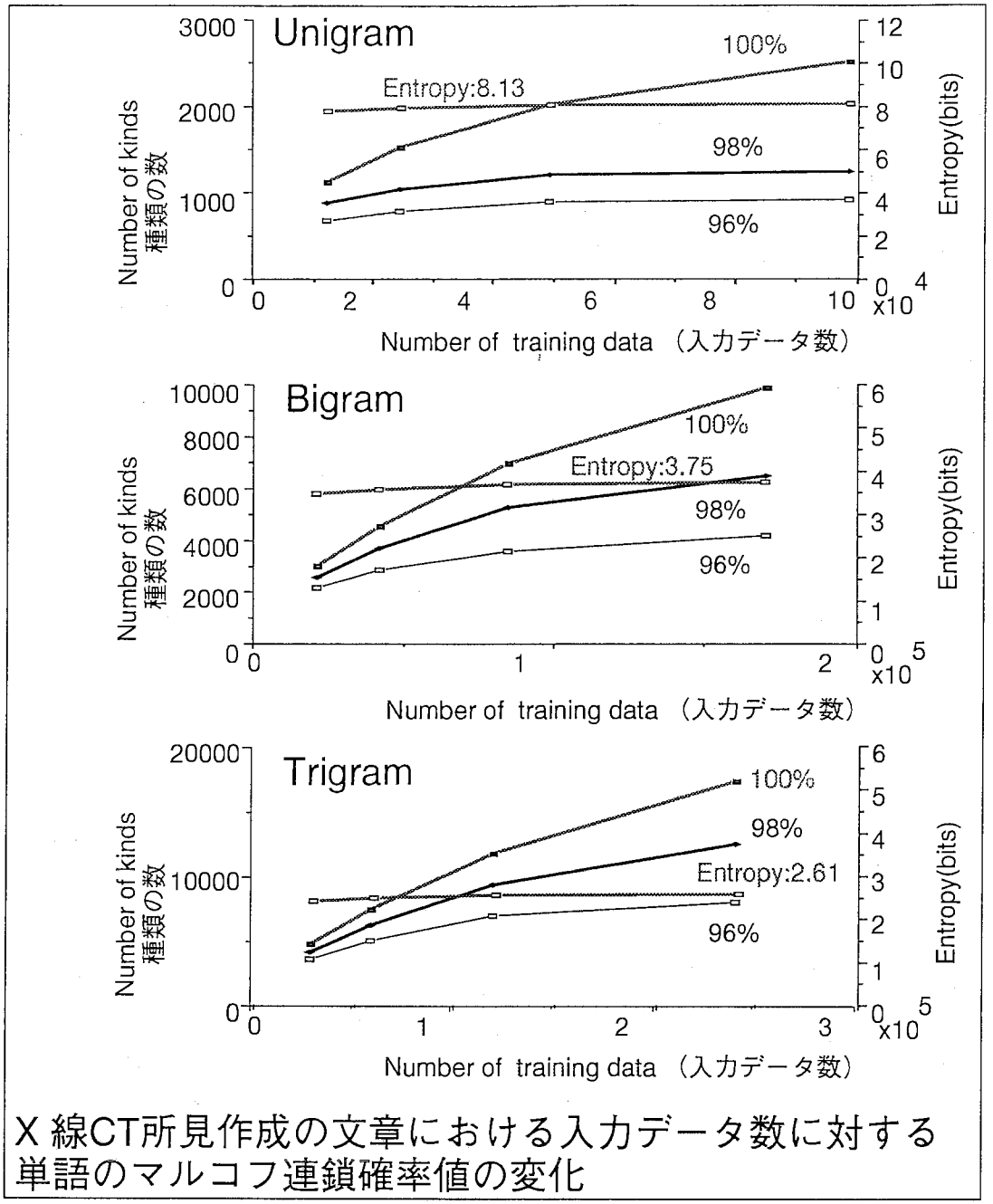


図 2.6: X 線 CT 所見における学習データ数に対する単語のマルコフ連鎖確率値の収束率

2.4 ATR の国際会議における単語 trigram の値の収束率

単語の trigram の値の信頼性を調べるために、ATR の国際会議の申し込みにおけるテキストデータベースにおいて、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。

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 2.7 に示す。

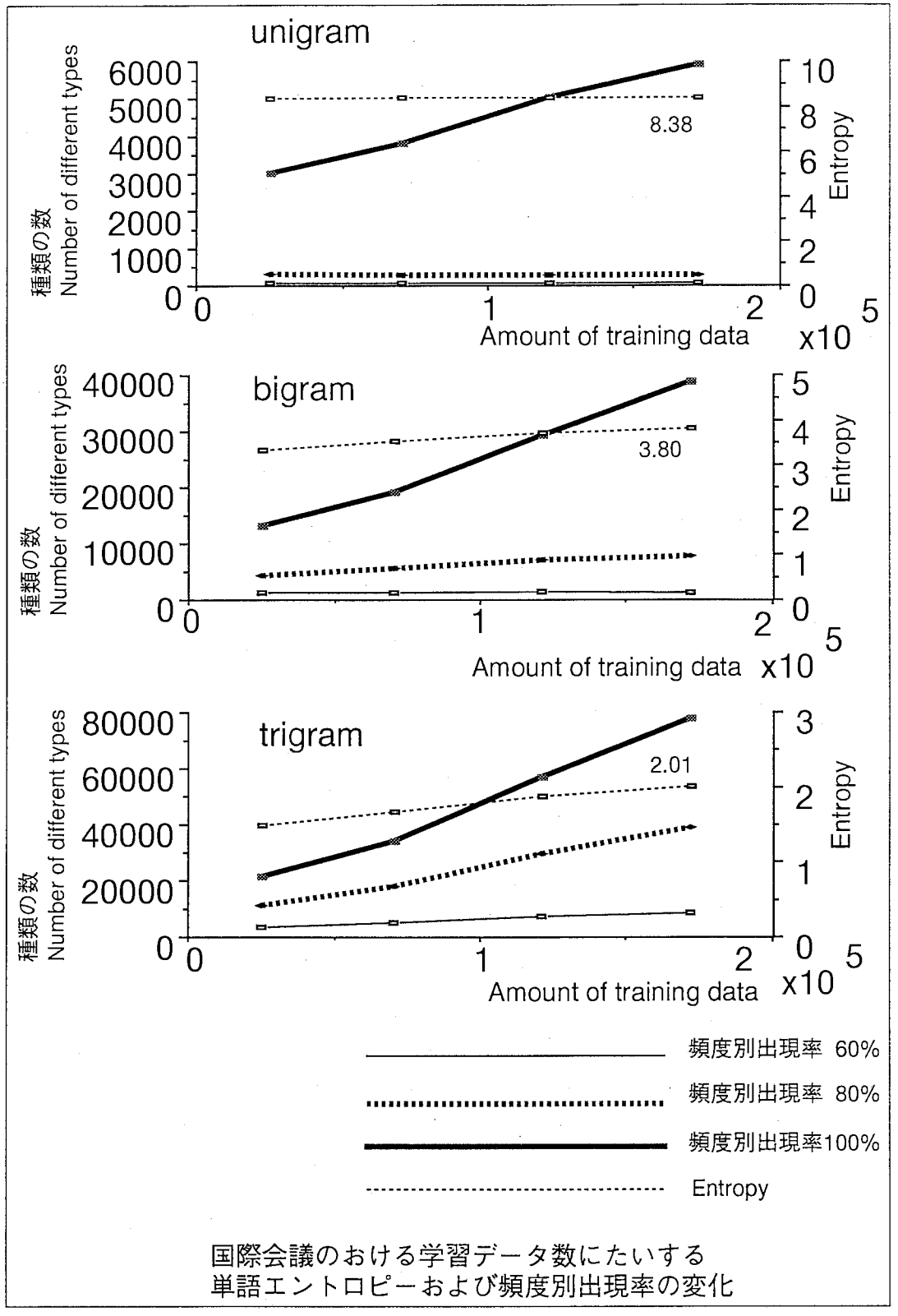


図 2.7: 学習データの入力データに対するエントロピーおよび頻度別出現率の変化
Entropy and coverage rate versus amount of training data

2.5 入力データ量に対するマルコフ連鎖確率値の収束率について

ここでは、新聞記事およびX線CT所見作成において、学習データ量に対する音声・漢字仮名・品詞・単語のマルコフ連鎖確率値の収束率について調べた。これらの結果から、以下のことが示される。

2.5.1 エントロピーと頻度別出現率

エントロピーと頻度別出現率の収束性を比較すると、全てのデータにおいてエントロピーは頻度別出現率よりも少ない学習データ量で収束することが示された。これは学習データ量に対するマルコフ連鎖確率値の変化について調査する場合、エントロピーだけでなく、頻度別出現率も調査する必要があることを意味していると思われる。

2.5.2 頻度別出現率100%と98%

頻度別出現率100%のデータと頻度別出現率98%のデータの比較から、学習データが増加した場合、全体に占める割合は少ないが、たえず新しい種類のマルコフモデルの組み合わせが出現することが予想される。これは、言語モデルとしてマルコフモデルを選択したときの妥当性に関して、2つの異なった解釈を出す。1つは、滅多に出現しない言語現象は、あえてモデルに適合させる必要がないと判断するもので、もう1つは、所詮、言語モデルとしてマルコフモデルは使用できないと判断するものである。この判断は、人によって異なると思われる。

2.5.3 新聞記事とX線CT所見作成の比較

X線CT所見作成の文章と新聞記事を比較すると、音節・漢字仮名、いずれの場合もエントロピーが低く、かつ少ない学習データ量で収束している。これらからX線CTの所見作成の文章は新聞記事と比較して文章が単純であると言える。

2.5.4 形態素解析プログラムの精度

新聞記事におけるマルコフ連鎖確率の収束性を調べるために使用した形態素解析プログラムは単語認定率で約95%の精度しかないため、人手によって文節単位に区切られた場合のマルコフ連鎖確率の値と、ここで得られた値に差がある可能性がある。特に品詞に関しては、品詞のtrigramに有意性が見られなかった。これは、品詞の定義が人によって異なる（例えば形容動詞）などの問題点もあるが、形態素解析の精度の問題と関連している可能性があり、今後検討が必要である。

2.5.5 ATRの国際会議における単語trigramの値の信頼性

図2.7から、データ量が増加するに伴いエントロピーは増加していて、安定な値になっていないことがわかる。また語彙の58.8%(3486/5933)、単語trigramの種類の数の77.9%(60847/78138)は1回しか出現していなかった。このデータをX線CT所見と比較すると単語のエントロピーの絶対値では差が少ないことがわかる。したがって、固有名詞など1度しか出現しない単語が多過ぎることを意味している。そしてデータ収集に問題があると考えている。

第 3 章

自由発話の音声的、言語的特徴

近年、連続音声認識の研究が盛んに行なわれ、多くの研究機関で文音声音声システムが構築されている [27],[41],[64],[50]。これらのシステムの多くは、朗読発話のような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」などの間投詞や、言い淀みや言い誤りおよび言い直しなどが頻繁に見受けられる。このような音声の認識が今後の重要な研究課題になると思われる。

この研究の第一歩として、本論文では視察によるラベリングをして自由発話の音声データを調べた。自由発話の定義は研究者によって異なるが、ここでは話者がテキストを見ないで対話した音声を自由発話と見なした。そして自由発話と朗読発話の差を見るために、間投詞と言い直しの出現頻度、発話速度、融合ラベルの付与率、HMM による認識精度などを調査した。だたし、音響的な傾向に関して調査した話者は 4 名のみであるため、調査結果の値は一般性に欠ける可能性がある。

なお、自由発話の視察によるラベリングには多くの人手が必要であるため、このような報告は少ない。文献 [25] において、小林らは日本音響学会のデータベース [18] を利用して、自由発話の文の中に出現するポーズの長さを報告している。一方、音声を文字化するコストはラベリングのコストよりも少なくてすむため、間投詞や言い直しなどの言語現象を調べた論文は比較的多い。日本語では、間投詞や言い直しの出現頻度を調べた報告 [36],[13],[4] や、助詞落ち・倒置の分析を行なった報告 [65] などがある。英語では自由発話のデータベースとして Air Travel Information Service (ATIS) がよく知られている。このデータベースを利用して自由発話の特徴を報告し [67],[68]、従来の音声認識で使用されたアルゴリズムを用いて、認識率を報告した論文が多く見られる [60],[69]。

3.1 自由発話の言語的な特徴

自由発話における言語の特徴については、自然言語処理の立場からどのような言語表現があるのかを調べた報告が既にある [4]。また、山本ら [65] は実際の対話文約 1800 文を名詞文節の助詞落ちや倒置の点から解析している。ここでは、自由発話と朗読発声の言語現象の差を調べる立場から、朗読発話では見られない言語（発話）現象、特に言い直しと間投詞に焦点をあてて、それぞれの出現頻度を調べた。今回調査した会話文は、国際会議の問い合わせの対話文 11054 文である。

3.1.1 調査に用いたデータベース

現在ＡＴＲでは、各種言語現象を調査するために対話文を中心とする言語データベースの作成を進めている [7]。本来、対話音声の収録は話者に録音していることを気づかれずに録音することが好ましいが、通信の守秘義務などの問題の他に、話題が次々に移行するため会話の語彙が

膨大な数になるという問題も生じる。このため、事前に話題のトピックやバックグラウンドを決め、会議の流れの不自然さを損なわないように打合せを行った後に収録をしている [53]。現在、発話内容で5種類、収録環境で2種類、話者で2種類、発話様式で2種類の variety を含むデータベースを収集中である [7]。このＡＴＲの言語データベースのなかから、申し込み者と事務局員が通訳を通して国際会議の問い合わせをしているデータを、自由発話の言語的な特徴を把握するための言語データベースとして利用した。このデータの収録条件を表 3.1 に示す。申し込み者役にはナレータ（アナウンサや声優など音声を職業としている人）の他に一般の話者も含まれているが、対応する事務局員役は、この分野の専門家が演じている。また収録は、遮音室の他に通常の部屋でも行なっている。

表 3.1: 調査に用いた自由発話の言語データ

発話内容	国際会議の申し込みに関する参加者と事務局の対話
データ量	3178 対話、 11054 文
発話様式	自由発話 「トピック」（質問項目と、その背景に関する情報）や「バックグラウンド」（会話の前提になる背景）を詳細に設定して対話したもの。
発話環境	
1 通常の部屋	大部分が家庭用のカセットテープレコーダで録音。外来雑音も混在。
2 スタジオ録音（遮音室）	ＤＡＴで録音。明瞭。
話者	
1 事務局員役	当該分野の専門家
2 申し込み者役	ナレータ ＋ 一般話者（複数話者）

3.1.2 自由発話の文例

自由発話では文の定義が曖昧になる。本論文では、発話権が相手から渡されて相手に返すまでの話者の発話を「文」と定義した。したがって、通常2文と考えられる文が1文になる。発話の例文を表3.2に示す。この例文において“『 』”で括られた範囲は1文を示している。また“『”の前の番号は、文番号である。

表 3.2: 文例

1 『［あっ、あの］わたくし、
武蔵野電機システム開発研究所の小金沢と申します。』
2 『［あのー］参加の申込みをしていたんですけれども、
［ちょっと］出られなくなりましたので、
［あの］キャンセルしたいんですが。』
3『はい。［あのー］キャンセルは書面にてっていうふうに
書いてあるんですけども、
どういうふうにしたらよろしいんでしょうか。』
4『［ああー］。』
5『［あっ］そうですか。』
6『［あっ］そうですか。』
7『それでですね、
［えーと］８０パーセント返していただけるのは
9月３０日までというふうに書いてあるんですけども、
［あのー］３０日までにそれが届けばですか。
そ　れとも、こちらが出した［その］消印が
３０日でも（かまい）かまわないんでしょうか。』
8『［ああ］そうですか。』
9『［あっ］そうですか。』
１０『わかりました。じゃ、登録ナンバーと名前、住所、所属なんかを
（かき）書いて、書いたものをお送りすればよろしいですね。』
１１『はい、どうも、ありがとうございました。』

3.1.3 自由発話の文の長さ

図 3.1に、自由発話における文の長さと出現回数を示す。自由発話においては「文」の明確な定義が困難である事例が多いため、文字化する際に意味的にまとまっていると判断できる単位を「文」とした。したがって文字に書き起こした人の主観のバラツキにより完全な統一はとれていない。（例えば「はい、もしもし」が１文になっていたり２文になっていたりする。）

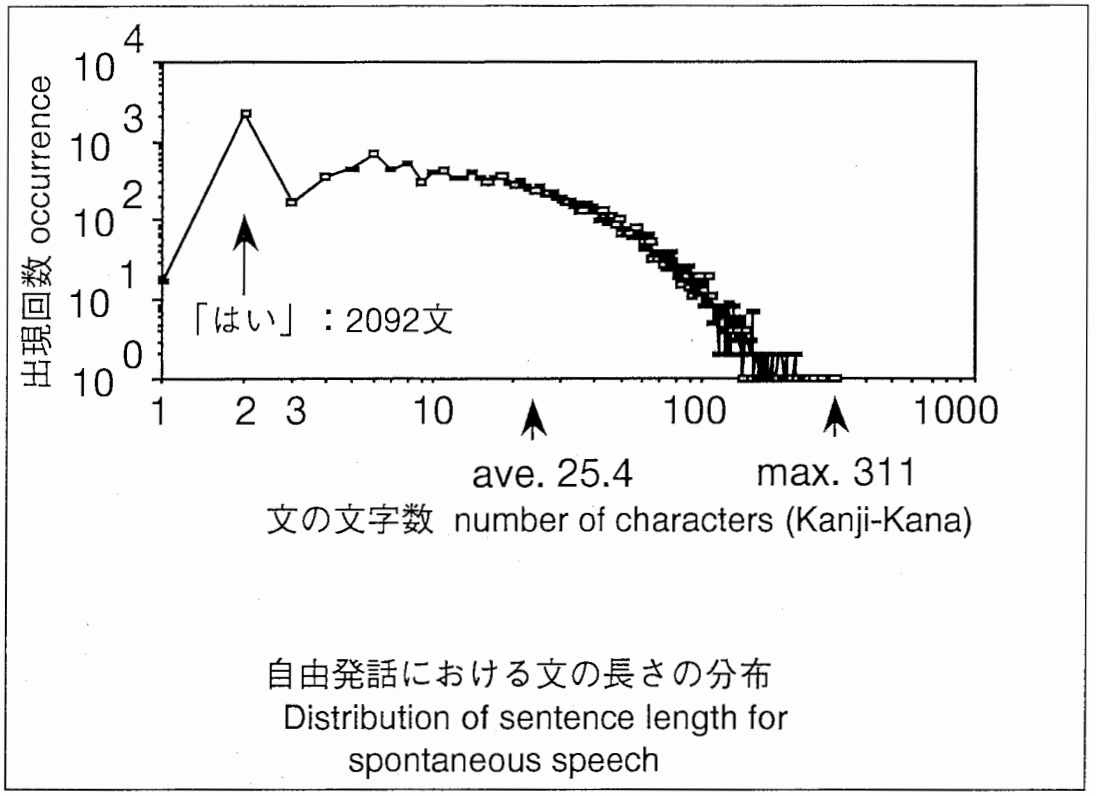


図 3.1: 自由発話における文の長さ

漢字仮名混じり表記に書き下したテキストデータを調査した結果、自由発話の 1 文の平均文字数は 25.4 文字、もっとも短い文は「あ」の 1 文字、もっとも長い文は 311 文字であった。最も出現頻度の高い文は、「はい」の 2092 文であった。また、自由発話の文の 73% は 32 文字以下であった。長い文の例を表 3.3に示す。

表 3.3: 長い文の例

えー、松下の場合にはですね、もうすでに、まー、あのー、見学コースっていうのが設定されておりまして、えー、会議の参加者のみならず、いろんな興味のある方々、これは日本人の方も外国人の方も見れる訳ですが、そういった、松下電器が、今までどの様な製品を作り、現在どの様なシステムで、えー、いろんな製品を作っておるか、そして、今後将来、松下がどういう方向性を目指してるか、という過去現在未来といった様な、製品の製作展開等のコースを見て頂くことになります。

3.1.4 自由発話における言い直しや間投詞の出現頻度

自由発話の音声には、「あのー」や「えーと」などの間投詞（冗長語）や、言葉の言い直しおよび言い誤りなどがある。これらの言語現象は、朗読発声では通常出現しないため、従来の文法の枠組では、あまり考慮されていない。そこで、自由発話における間投詞や言い直しの出現頻度を調べた。

なお、本論文では間投詞を『自立語。活用しない。主語・述語によらない。言い淀む場合などに、文の中に挿入されて用いられる。間投詞を取り除いても文の文法性および意味には影響しない [66]。』と定義した。また言い直しを『前にいった事の誤りを訂正してもう一度言う。』もしくは『他の適当なやさしい言葉で言う。』と定義した。また言い淀みを『言おうとしてためらったり、話しの途中でちょっと言葉につまったりする。』[52] と定義した。なお表 3.2の例文において、“[ ]” で括られた個所は間投詞、“( )” で括られた個所は言い直しを意味している。

この結果を図 3.2に示す。この結果において、間投詞も言い直しも共にない文は、全体の約 5 割であった。これらの多くは「はい」(23%) 「もしもし」(5%)「はい、わかりました」(3%)「どうも、ありがとうございました」(1.5%) などの定型文で、この種類の 8 割の文は 14 文字以下の短い文であった。

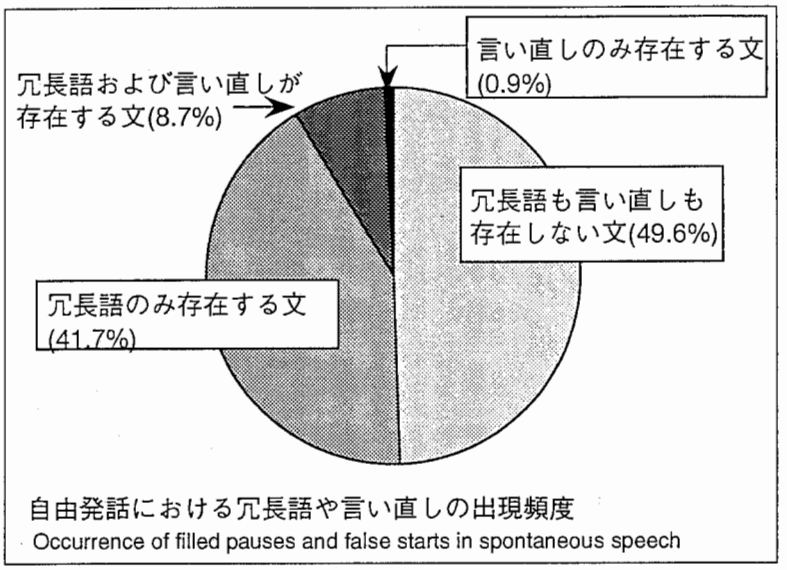


図 3.2: 自由発話における間投詞や言い直しの出現頻度

自由発話の文の約 5 割は間投詞を含み、多くの単語が続く文の多くは間投詞を含んでいた。ただし、間投詞には個人差が多く、間投詞を多く話す話者とあまり話さない話者がいた。また、一人の話者が話す間投詞の種類は限られていた。言い直しがある文は自由発話全体の約 1 割であった。そして、「はい」「もしもし」などの独立語も間投詞に含めた場合、全体の文の 83%(9121 文) は間投詞があった。この中で文頭に間投詞があるものは、全体の文の 65.8%(7303 文) であった。また、言い直しがある文の 46.1% は、言い直しの前もしくは後に間投詞が付加されていた。(言い直しの前に間投詞が付加されているのが 14%、言い直しの後に間投詞が付加されているのが 24%、言い直しの前後ともに間投詞が付加されているのが 8% であった。)

なお、今回調査した自由発話のデータは、ナレータや実際の事務局員など、言葉の対応に慣れた話者が発話した音声である。したがって、言葉の応対に慣れていない一般の話者では、間投詞や言い直しの出現頻度が増加する可能性がある。

3.1.5 自由発話における間投詞の種類と出現確率

自由発話において観測された間投詞の種類のなかで、出現頻度の高いものを図 3.3に示す。また観測された間投詞を表 3.4に示す。この表から、間投詞の種類はかなり多いが、上位 4 種類で間投詞全体の出現頻度の約 7 割を占めていることがわかる。

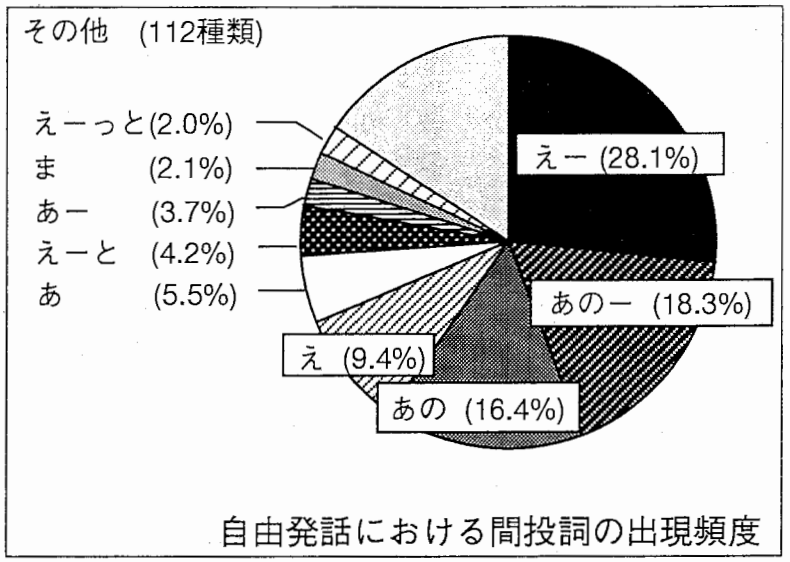


図 3.3: 自由発話における間投詞の種類と出現頻度

表 3.4: 自由発話における間投詞の一覧

間投詞	出現回数	間投詞	出現回数	間投詞	出現回数
「あ」	604	「えーっとお」	1	「その」	115
「あー」	268	「えーっとですね」	8	「そのー」	48
「あーっと」	2	「えーと」	466	「だか」	1
「あーと」	1	「えーとー」	4	「ちょっと」	8
「あーん」	5	「えーとですね」	3	「つ」	2
「ああ」	7	「えーまあ」	3	「で」	61
「あっ」	151	「えーん」	2	「でー」	13
「あっと」	1	「ええ」	13	「でぃ」	1
「あと」	1	「ええー」	1	「と」	77
「あなー」	1	「ええっと」	1	「とー」	11
「あの」	1809	「えっ」	22	「ねー」	1
「あのー」	2025	「えっーと」	4	「のー」	1
「あのーえー」	1	「えっと」	62	「は」	4
「あのう」	77	「えっとー」	11	「はあ」	1
「あのうー」	3	「えっとおー」	1	「はあー」	2
「あのと」	1	「えと」	47	「ははぁーん」	1
「あれ」	1	「えとー」	13	「ひ」	1
「あん」	1	「えへへっ」	1	「ふーん」	2
「い」	26	「えん」	1	「ま」	263
「いー」	58	「お」	59	「まー」	8
「いゃー」	1	「おー」	196	「まぁ」	186
「いやー」	2	「おーえー」	1	「まぁね」	1
「う」	23	「おっ」	2	「まあ」	176
「うー」	71	「ぐっ」	1	「まああのう」	1
「うーん」	26	「こう」	9	「まあまあ」	1
「うーんと」	2	「この」	9	「まっ」	5
「うっ」	1	「このー」	4	「も」	2
「うん」	7	「じゃ」	4	「もう」	1
「え」	1040	「じゃー」	1	「よ」	1
「えー」	3105	「じゃあ」	1	「りー」	1
「えーえ」	1	「す」	8	「わあ」	1
「えーちょっと	1	「すー」	2	「わっ」	1
「えーっ」	1	「すい」	1	「ん」	27
「えーって」	1	「すっ」	2	「んー」	19
「えーっと」	256	「せ」	1	「んっ」	1
「えーっとー」	2	「そ」	2	「んっと」	1
「えーっとえー」	1	「そう」	1	「んで」	1
				「んと」	2

自由発話中では、しばしば間投詞と言い淀みの区分が不明確になる。例えば「100 パーセント、え 日本語と英語で行われます。」における「え」は、間投詞とも「英語」の言い淀みとも

解釈できる。また、語尾音の継続時間には、話者間に大きなバラツキがあるため、語尾の伸びる語と伸びない語（例えば「えー」と「え」）の決定は、文字化した人の判断に依存している。ここで示した間投詞の出現頻度のデータには、このような意味で曖昧さがある。

なお、話し相手と対面して話す自由発話に対して、電話のような音声のみによる対話では、間投詞は相手の注意を促す役割を持つ場合がある [13]。このため、今回調査した間投詞の出現頻度は、高めに評価されている可能性がある。

また、自由発話における間投詞の出現頻度は多くの研究期間で報告されている。文献 [25] や文献 [55] では日本音響学会連続音声データベースの書き起こしテキストを調査して報告している。また、文献 [13] では、本論文で使用したデータベースの小量のときの開始符合の種類を示している。この場合、「えー」、「えーっ」などの単語が多いことを報告している。また、文献 [48] では NHK ラジオ第一放送の電話相談番組を書き起こしている。これらの報告と比較すると比率に違いがあるが、代表的な間投詞に関してはほぼ同じ割合といえる。

3.1.6 自由発話における言い直しの種類と出現頻度

自由発話において特有な言い誤りは、文法的、意味的な前後関係を考慮して決定する必要がある。また、言い淀みは音声を注意深く聞いて決定する必要がある。したがって言い誤りや言い淀みの言語現象は話者が言い直さないかぎり検出するのは困難である。したがって本論文では言い直しの出現頻度のみを調査した。調査は 200 文に対して行なった。この言い直しの分類と出現頻度を、図 3.4に示す。また例文を以下に示めす。例文中においてアンダーラインは言い直しを意味する。

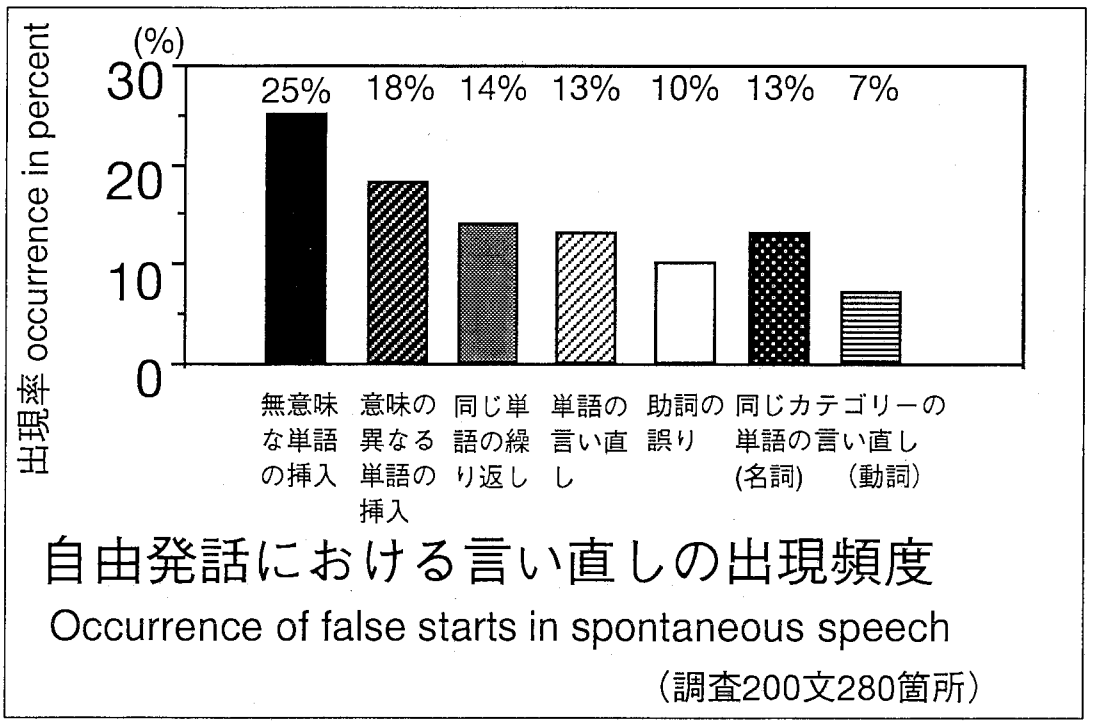


図 3.4: 言い直しの出現頻度

自由発話における言い直しの例文

1. 無意味な単語の挿入　25%

- 日本語から英語へというように
、と、翻訳を、す、あ、通訳をするコンピュータを開発している
（「通訳」と言おうとして「翻訳」と言い間違いをし、これに気がついて直そうとして言い淀んでいる。）
- えーっと、あのー、こ、会議期間中は特にあのー、バスを運行しておりまして、土曜ダイヤでバスが、あのー、運行するようになっております。
（原因が不明、「こ」は、無意味な音の発声であるため、間投詞と判断される可能性がある。）
- 最終的な、えーっと、草稿、原、えーとスピーチ原稿を提出していただきたいと思います。
（「原稿」と言おうとして「草稿」と言い間違いをし、これに気がついて直そうとして言い淀んでいる。）
- パンフレットの方を拝、見ていただきましたら
（「拝見」と言おうとして敬語の間違いをして言い淀んでいる。）

2. 意味の異なる単語の挿入　18%

- あの、そのようなことが、あの、そちらの方にお教え、お知らせできないんです。
（「知らせる」を「教える」に言い間違えている。）
- タクシーに、あのー、京都駅からお乗りになれば、大体３５分か４０分位で着きますし、旅費、料金としては、大体１５００円位になります。
（「旅費」と「料金」は、意味的にはほとんど同じであるため、『丁寧な言葉への言い直し』とも分類できる。）
- この件に関しましては、えーっと、大阪まで、あのー、新幹線で来られますと、飛行機で来られますと４５分間位で参ります。
（「飛行機」を「新幹線」と言い間違えている。文全体の挿入の誤り。）

3. 同じ単語の繰り返し　14%

- えーっと、その、その中でちょっと、あの、クレジットカードをね書類の方は、
（「その中」を１つの単語と捉えたならば『単語の言い直し』とも解釈できる。）
- 会議の内容なんかをかいつまんでお話、お話し下さればと思うんですが。

4. 単語の言い直し　13%

- あの、この、クレ、クレジットカードというのは本来外国人のゲストの方
- 従いまして、２、あ、２時間半位で東京から国際会議の行なわれる場所まで行けるわけですから、

5. 助詞の誤り　10%

- まだ割引を私の方で、あのー、することに、はできないんですが
- はい、それで、はそうですね。
- コンピュータによる同時通訳を、に関する、あのー会議を開こうということです。
- オーバーヘッドープロジェクタと２インチ×２インチのスライドを、と使えるようになっています。

6. 丁寧な単語を用いた言い直し（名詞） 13%

- えーっと、郵送でＶＬＤＢ８６の、えーと、会議事務局、国際会議事務局宛にお送りいただきたいと思います。
  （意味的には『同じ単語の繰り返し』ともみなせる。）
- それで、えーっと、受領の通知は、受け取りの通知は１２月３１日までに出させていただきます。
- これは現在の為替でいきますと、レートでいきますと、大体16,000円程になりますので
- その次に日本の総理大臣中曽根首相から挨拶を、スピーチをすることになってます。

7. 丁寧な単語を用いた言い直し（動詞） 7%

- はがきででも来られない、参加できないという風に、御通知いただければ、
- ええ、外国人の申し込みの方は、現在までで１３名であり、ございます
- そうですか、という、といいますと、それは英語でしなければいけないわけでしょうか。

この図における言い直しの分類は、かなり主観的である。例えば、単語の意味の違いは明確でないため『意味の異なる単語の挿入』と『丁寧な単語での言い直し』の区別の差は明確でない。また、日本語では単語の概念が曖昧なため、『同じ単語の繰り返し』と『単語の言い直し』の区別の差も明確でない。

なお文献[54]では言い直した単語に着目して、言い直した単語の長さを報告している。これを見ると言い直しの59%は、言い誤った単語を直ちに言い直している。この傾向はほぼ同じである。また文献[45]においてもほぼ同様な結果が見られる。この論文では単語にならないsyllaleが39%、直後に言い直しているのが52%であることが示されている。これらの結果は、今回の結果に類似している。

3.2 各話者における自由発話の言語的な特徴

自由発話における個人差を見るために4名の話者を個別に調査した。この音声データは、ナレータ（アナウンサや声優など音声を職業としている人）が申し込み者の役になって発話しているため、舌打ちの音などはほとんどない。事務局員側とは完全に分離されて録音されているため音声区間の重畳はない。収録は遮音室で行なわれたためドアの開閉音などの日常雑音はない。したがって、この音声データは自由発話としてはかなりcleanな音声であると言ってよい。このデータの収録条件を表3.5に示す。ただし、各々の話者の発話内容は異なっている。

表 3.5: 調査に用いた自由発話の音声データの収録条件

話者	ナレータ 4 名
収録環境	遮音室
発話内容	国際会議の申し込みに関する参加者と事務局の対話
発話様式	自由発話 「トピック」（質問項目と、その背景に関する情報）や「バックグラウンド」（会話の前提になる背景）を詳細に設定して対話したもの。
入力系	マイクロフォン、ＤＡＴ録音
データ量	13 対話 116 文 3943 音素 (MTK) 18 対話 333 文 11520 音素 (MMY) 13 対話 180 文 6918 音素 (FKN) 14 対話 195 文 7588 音素 (FAK)

また、自由発話の音声データを文字に書き起こした後に、間投詞や言い直しを削除して作成したテキストを、自由発話と同一の話者が発声した音声を、朗読発話の音声データとして使用した。したがって、同一話者における自由発話と朗読発話の発話内容は、間投詞および言い直しを除いてほぼ同一である。

3.2.1 話者ごとの間投詞や言い直しの出現頻度

自由発話には、「あのー」や「えーと」などの間投詞（冗長語）や、言い直しがある。これらの出現頻度を各話者ごとに調べた。各話者ごとの間投詞の出現頻度を図 3.5に、言い直しの出現頻度を図 3.6に示す。

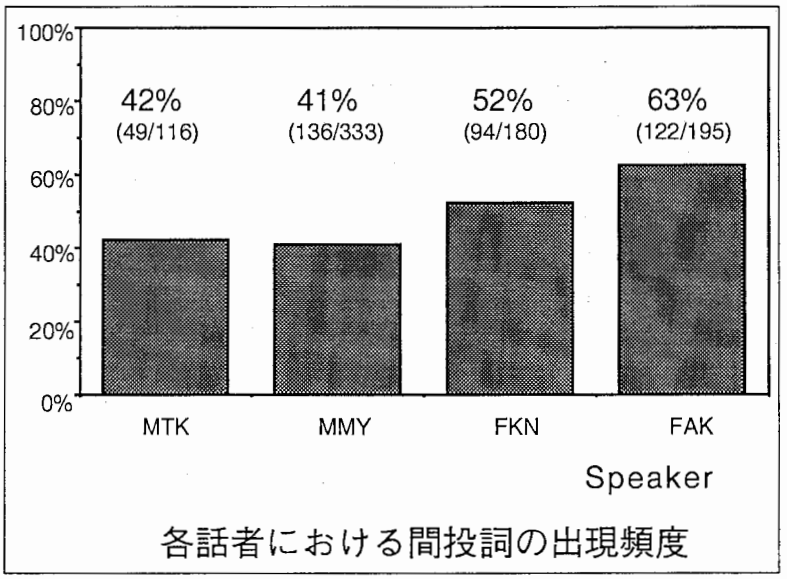


図 3.5: 各話者における間投詞の出現頻度

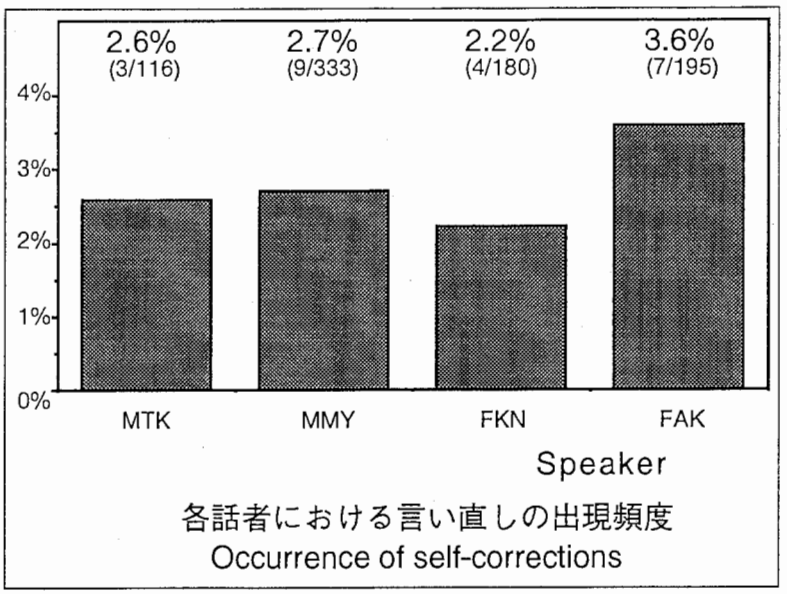


図 3.6: 各話者における言い直しの出現頻度

これらから次のようなことが判る。

1. 間投詞は、話者によって相違が見られるが、文章全体の 40% から 60% の文に出現する。

2. 言い直しは、話者によって相違が見られるが、文章全体の 2% から 4% の文に出現する。

3.2.2 話者ごとの間投詞の種類や出現頻度

間投詞には多くの種類があるが、出現頻度の高い間投詞は限られていることが既に報告されている [36],[24],[48]。ここでは各話者ごとに、出現頻度の高い上位 4 つの間投詞の種類と、その出現頻度を調べた。この結果を図 3.7に示す。

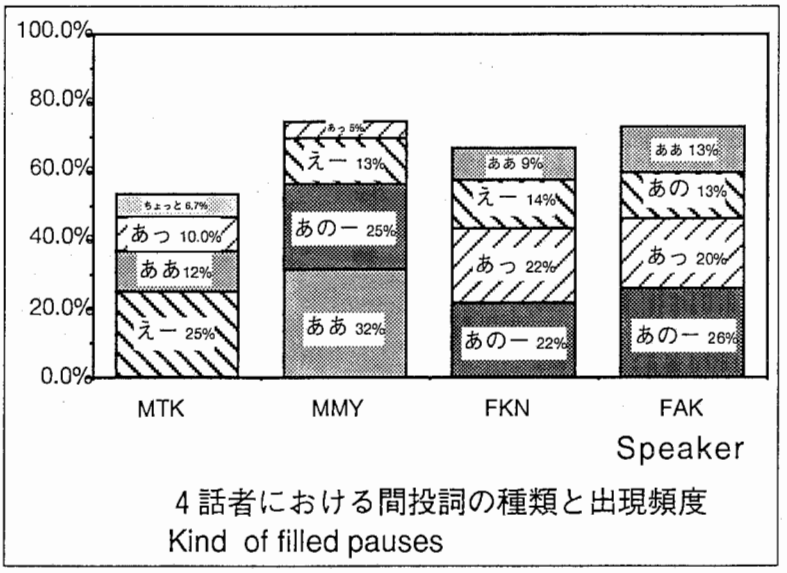


図 3.7: 間投詞の種類別の出現頻度

これらから次のようなことが判る。

1. 間投詞全体の 50% から 75% は、使用頻度の高い 4 種類の間投詞で占める。

2. 使用頻度の高い間投詞の種類は、話者によって相違がある。

3.3 各話者における自由発話の音響的な特徴

3.3.1 ラベリング作業からみた自由発話

音声データのラベリングの作業において見受けられた自由発話の音素の定性的な特徴を表 3.6に示す。なお、ラベリングの基準は文献 [56] に従った。これらから自由発話では音素境界がかなり曖昧になっていることや、従来の朗読発話には見られない音素が現れていることがわかる。

表 3.6: 自由発話の音素の定性的特徴

1	文の語尾の音素が不明瞭になることもある。 （例: 「なんですか」の「か」がほとんど聞こえない。）
2	母音 /a,i,u,e,o/ 全てが無声化することもある。 （朗読発話では /a,e,o/ は、あまり無声化しない。）
3	2重に解釈できる音素がある。 （例: 「んー」（考え込むとき発声している音）は /N/ あるいは /uN/ の 両者に解釈できる。）
4	子音 /r/ をともなう音節の発音が全体的に弱い。 （例: 「そうする と」の「る」がほとんど聞こえない。）
5	母音（特に、文末の母音『a』）の第1フォルマントが あらわれないことがある。

3.3.2 各話者における融合ラベルの付与率

ＡＴＲでは、発話テキストを参照しながら人手で音素境界を決定するラベリング作業において、音素境界が不明瞭な音素区間に対して付与するラベルのことを融合ラベルと呼んでいる。この融合ラベルの付与率を 4 名の話者の自由発話と朗読発話において調査した。この結果を図 3.8に示す。この図から読みとれることを以下に示す。

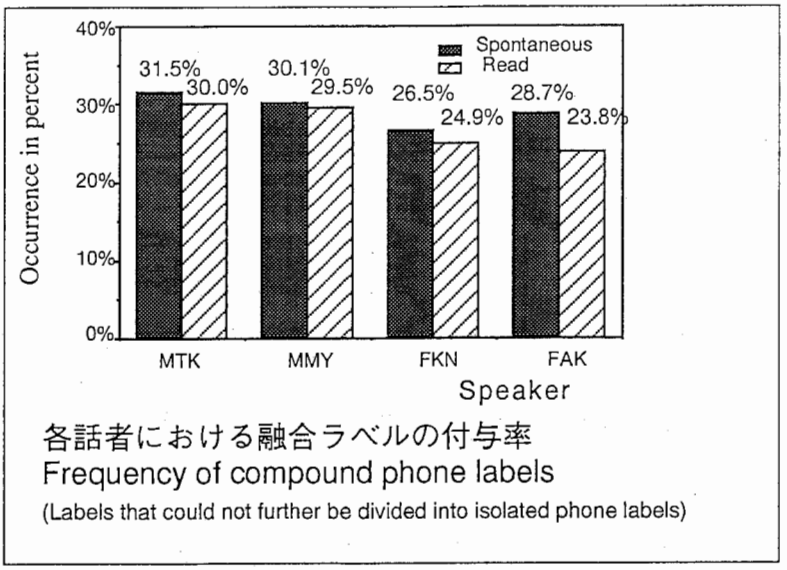


図 3.8: 話者ごとの融合ラベルの付与率の変化

1. 自由発話の融合ラベルの付与率は朗読発話より高い。
2. 自由発話、朗読発話共に、融合ラベルの付与率に話者の相違が見られる。
3. 自由発話では、全音素の 25% から 32% が融合ラベルになる。
4. 朗読発話では、全音素の 24% から 30% が融合ラベルになる。
5. 自由発話と朗読発話を比較すると、融合ラベルの付与率の増加の割合に話者の相違が見られる。話者 MMY では 2%(29.5% → 30.1%) しか増加しないのに対し、話者 FAK では 21%(23.8% → 28.7%) 増加する。

3.3.3 各話者における発話速度の違い

ここでは、自由発話と朗読発話の発話速度の差をモーラ速度で調査した。ただし、息つぎなどの長いポーズ区間および間投詞および言い直しの音声区間は除去した。この結果を図 3.9に示す。

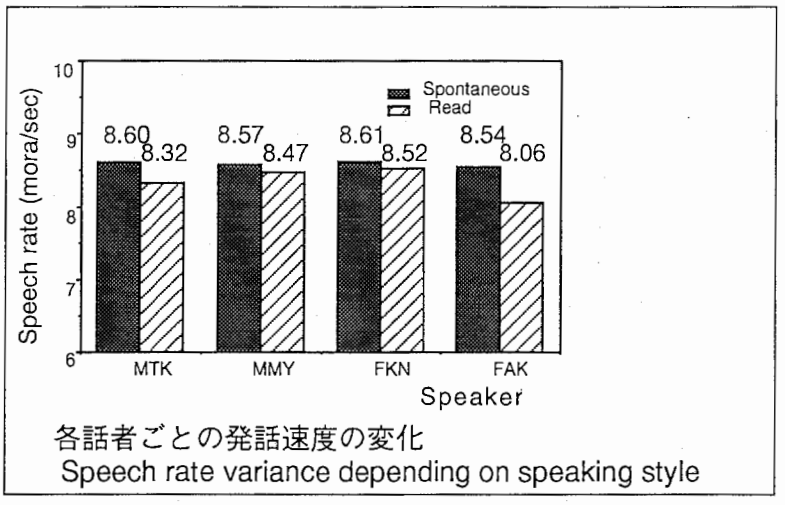


図 3.9: 各話者における発話速度の変化

また、自由発話および朗読発話における各音素の平均音素継続時間を表 3.7に示す。ただし融合ラベルが付与された音素は評価対象から削除した。

表 3.7: 各音素ごとの平均音素継続時間

自由発話

話者	MTK		MMY		FKN		FAK	
音素	平均 (ms)	標準偏差	平均 (ms)	標準偏差	平均 (ms)	標準偏差	平均 (ms)	標準偏差
a	92.6	49.1	87.6	41.0	86.1	38.2	93.4	63.4
i	72.0	36.1	65.1	39.1	64.2	36.8	70.5	45.5
u	85.0	62.0	77.0	53.8	69.0	43.4	77.7	45.5
e	93.5	64.3	92.7	72.2	80.6	44.3	85.0	53.7
o	91.8	61.5	97.4	71.6	92.5	59.6	101.3	74.5
p	61.4	7.1	17.4	6.9	20.7	7.8	12.8	4.5
t	41.4	21.0	20.4	7.1	17.2	5.7	14.8	5.8
k	49.3	19.9	36.3	19.3	30.2	10.5	26.6	12.1
b	48.3	13.4	11.1	5.0	11.8	4.4	11.7	2.4
d	43.5	16.9	12.0	4.7	11.6	4.0	10.8	3.3
g	45.9	25.3	13.8	4.7	13.9	5.5	14.3	5.2
m	53.1	17.3	47.1	19.6	52.6	15.6	51.8	17.2
n	48.2	18.3	41.6	20.0	47.9	19.1	53.4	24.6
N	75.7	24.0	66.6	37.5	55.2	28.2	70.7	32.4

朗読発話

話者	MTK		MMY		FKN		FAK	
音素	平均 (ms)	標準偏差	平均 (ms)	標準偏差	平均 (ms)	標準偏差	平均 (ms)	標準偏差
a	100.2	57.5	88.7	36.3	88.0	33.3	91.5	43.1
i	81.3	46.9	70.8	36.3	63.6	30.9	70.7	35.1
u	76.1	48.6	71.2	42.0	66.8	39.4	73.3	38.5
e	92.2	67.3	85.6	58.6	73.6	28.6	82.3	41.8
o	86.9	50.2	91.7	53.5	86.5	38.5	94.5	46.7
p	61.3	14.1	17.6	7.4	16.8	4.5	17.9	8.9
t	40.1	22.9	18.6	6.7	15.3	4.6	13.8	4.3
k	42.5	18.7	35.0	18.1	28.1	10.5	28.1	12.8
b	52.9	38.9	9.7	2.2	11.6	5.1	10.0	0.0
d	43.1	18.7	11.7	4.1	11.3	3.1	10.0	2.5
g	45.6	26.7	19.7	9.8	13.6	5.7	13.5	4.0
m	53.4	18.1	48.1	19.0	47.2	14.2	50.1	14.6
n	47.3	15.5	44.5	21.7	44.1	17.6	48.4	16.1
N	103.6	57.4	70.2	33.4	63.3	37.7	65.3	24.1

これらの結果から読み取れることを以下に示す。

1. 自由発話の発話速度は朗読発話より早い。

2. 自由発話の発話速度は 8.5(mora/sec) から 8.6(mora/sec) である。

3. 朗読発話の発話速度は自由発話より話者の相違が大きく 8.1(mora/sec) から 8.5(mora/sec) である。

4. 自由発話と朗読発話を比較すると、発話速度の増加の割合に話者の相違が見られる。話者 MMY では 1.2% しか増加しないのに対し (8.47 → 8.57)、話者 FAK では 6.0%(8.06 → 8.54) 増加する。

5. 話者 FAK を除くと、自由発話における母音 /a/ の平均音素継続時間は朗読発話より短い。しかし母音 /u/,/e/,/o/ は朗読発話より長い。

6. 話者 MTK を除くと、多くの音素では自由発話の音素継続時間の分散は朗読発話より大きい。

3.3.4 認識精度 (Phone Accuracy) から見た自由発話

ここでは自由発話と朗読発話の差を、連続音素認識実験をして正解率 (Phone Correct) および認識精度 (Phone Accuracy)[44],[10] で評価した。特定話者の同一発話様式の認識実験を行なうために、同一話者の同一発話様式の音声データの、文番号の奇数番目を学習データに偶数番目を評価データにした。学習プログラムには主に HTK Software Tools[10] を使用した。特徴パラメータには LPC ケプストラムを使用し、HMM には混合連続分布型を用いた。表 3.8 に実験条件を示す。

表 3.8: 音素認識の実験条件

認識対象	26 音素
サンプリング周波数	12kHz
話者	男性のナレータ
学習データ	同一発話様式
音響パラメータ	log power + 16 次 LPCcepstrum
	+ Δlog power + 16 次 Δcepstrum
フレーム窓長	20ms
フレーム周期	5ms
LPC 分析	16 次
打ち切り次数	16 次
音素モデル	4-state 3-loop 3 mixture
	Gaussian continuous HMM (diagoal)

認識実験は以下のようにしておこなった。

1. 学習データにおいて、融合ラベルが付与されなかった音素のみを切り出して Baum-Welch 学習をする。学習回数は１０回。

2. 学習データを文単位で連結学習する。学習データは間投詞や言い直しを含む。学習回数は３回。

3. 学習データと同一話者・同一発話様式の評価データを文単位で連続音素認識する。なお評価データは間投詞や言い直しを含む。

4. 評価データの音素ラベルを正解として、音素正解率 (Phone Correct) と音素認識精度 (Phone Accuracy) を計算する。

3.3.5 音素認識実験から見た自由発話

図 3.10に、認識実験の結果得られた音素正解率 (Phone Correct) と音素認識精度 (Phone Accuracy) を示す。また母音の音素認識誤り傾向を表 3.9に示す。

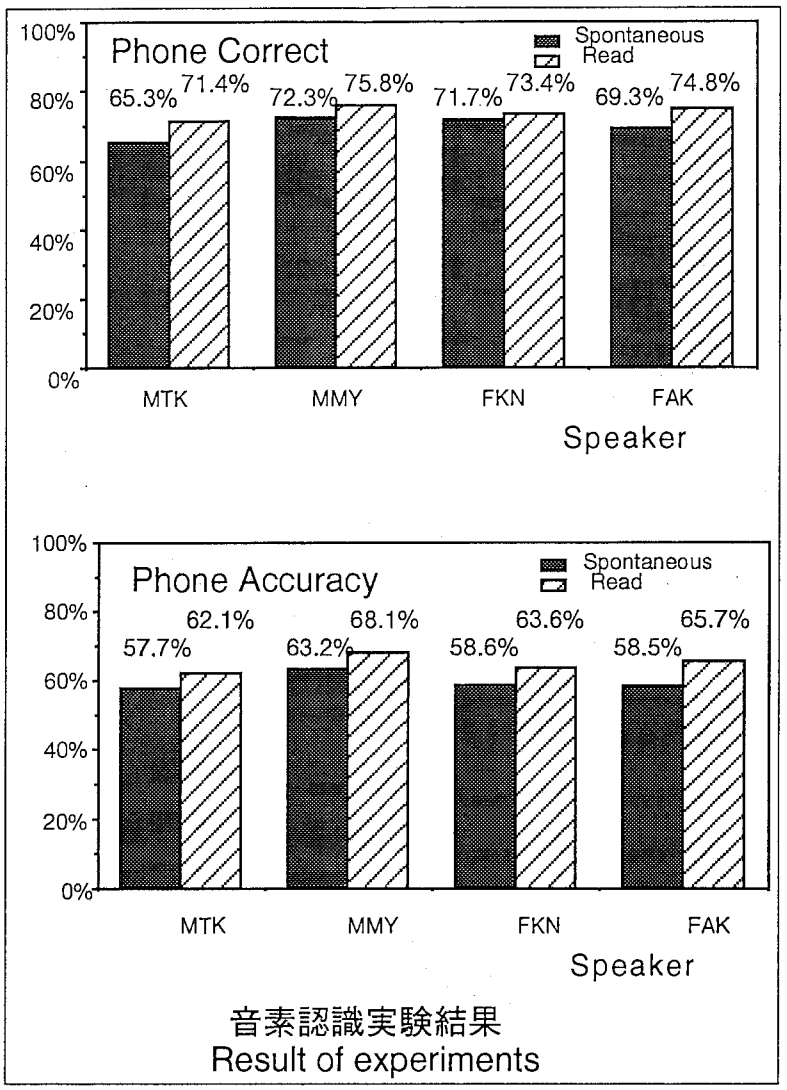


図 3.10: 音素認識率 (%)

表 3.9: 音素認識誤り傾向

話者 MTK (認識音素数 / 調査音素数)

		出力				
		a	i	u	e	o
入力	a	83.1% (167/201)	0.0% (0/201)	1.5% (3/201)	3% (6/201)	7.5% (15/201)
	i	0.7% (1/128)	85.1% (109/128)	3.9% (5/128)	3.9% (5/128)	0.7% (1/128)
	u	7.3% (6/ 82)	4.8% (4/ 82)	48.7% (40/ 82)	2.4% (2/ 82)	9.7% (8/ 82)
	e	3.0% (4/131)	13.7% (18/131)	1.5% (2/131)	76.3% (100/131)	2.2% (3/131)
	o	3.5% (5/140)	0.7% (1/140)	5.7% (8/140)	2.8% (4/140)	80.0% (112/140)

話者 MMY (認識音素数 / 調査音素数)

		出力				
		a	i	u	e	o
入力	a	93.2% (633/679)	0.1% (1/679)	0.7% (5/679)	3.0% (21/679)	1.1% (8/679)
	i	0.0% (0/426)	81.4% (347/426)	3.2% (14/426)	4.9% (21/426)	0.0% (0/426)
	u	1.2% (4/320)	4.0% (13/320)	45.6% (146/320)	3.4% (11/320)	7.1% (23/320)
	e	1.4% (6/405)	3.4% (14/405)	2.2% (9/405)	83.4% (338/405)	0.7% (3/405)
	o	1.5% (8/522)	0.0% (0/522)	1.7% (9/522)	3.4% (18/522)	88.5% (462/522)

話者 FKN (認識音素数 / 調査音素数)

		出力				
		a	i	u	e	o
入力	a	83.7% (381/455)	0.4% (2/455)	1.9% (9/455)	4.6% (21/455)	1.5% (7/455)
	i	0.0% (0/289)	76.4% (221/289)	2.0% (6/289)	3.8% (11/289)	0.3% (1/289)
	u	1.4% (3/205)	0.9% (2/205)	52.6% (108/205)	9.7% (20/205)	4.3% (9/205)
	e	0.4% (1/227)	4.8% (11/227)	3.0% (7/227)	84.1% (191/227)	0.0% (0/227)
	o	1.2% (4/318)	0.0% (0/318)	4.4% (14/318)	0.3% (1/318)	88.6% (282/318)

話者 FAK (認識音素数 / 調査音素数)

		出力				
		a	i	u	e	o
入力	a	80.6% (393/487)	0.0% (0/487)	4.1% (20/487)	4.7% (23/487)	2.0% (10/487)
	i	0.0% (0/265)	73.9% (196/265)	1.1% (3/265)	7.9% (21/265)	0.3% (1/265)
	u	6.0% (12/199)	3.5% (7/199)	43.2% (86/199)	6.0% (12/199)	4.0% (8/199)
	e	0.8% (2/244)	9.0% (22/244)	2.8% (7/244)	78.6% (192/244)	0.4% (1/244)
	o	2.6% (10/381)	0.0% (0/381)	3.9% (15/381)	1.5% (6/381)	83.7% (319/381)

これから次のような結果が示される。

1. 自由発話は朗読発話と比較して、音素正解率も音素認識率も低下する。
2. 自由発話の正解率 (Phone Correct) は、65% から 72% が得られた。
3. 自由発話の認識精度 (Phone Accuracy) は、58% から 63% が得られた。
4. 自由発話は朗読発話と比較すると認識精度は 7% から 10% 程度低下する。

5. 各音素の認識率をみると、母音の /u/ の認識精度が他の音素と比較して低い。

3.4 発話様式から見た自由発話

ここでは話者2名において単語発話、文節の朗読発話、文の朗読発話、自由発話における融合ラベルの付与率、発話速度、および音素認識誤り率を調べた。ただし、文節の朗読発話と文の朗読発話の発話内容は同一であるが、単語発話および朗読発話および自由発話の発話内容は異なる。また、単語発話、文節の朗読発話、文の朗読発話の発話内容は話者間に相違はないが、自由発話では、各話者の発話内容は異なっている。なお単語発話のデータは通称 (D0-D5)、文節の朗読発話は通称 DSA、文の朗読発話には通称 DSC と呼ばれているものを使用した。

3.4.1 融合ラベルの付与率から見た自由発話

融合ラベルの付与率を、同一話者の4種類の発話様式において調査した。この結果を図 3.11に示す。この図から読みとれることを以下に示す。

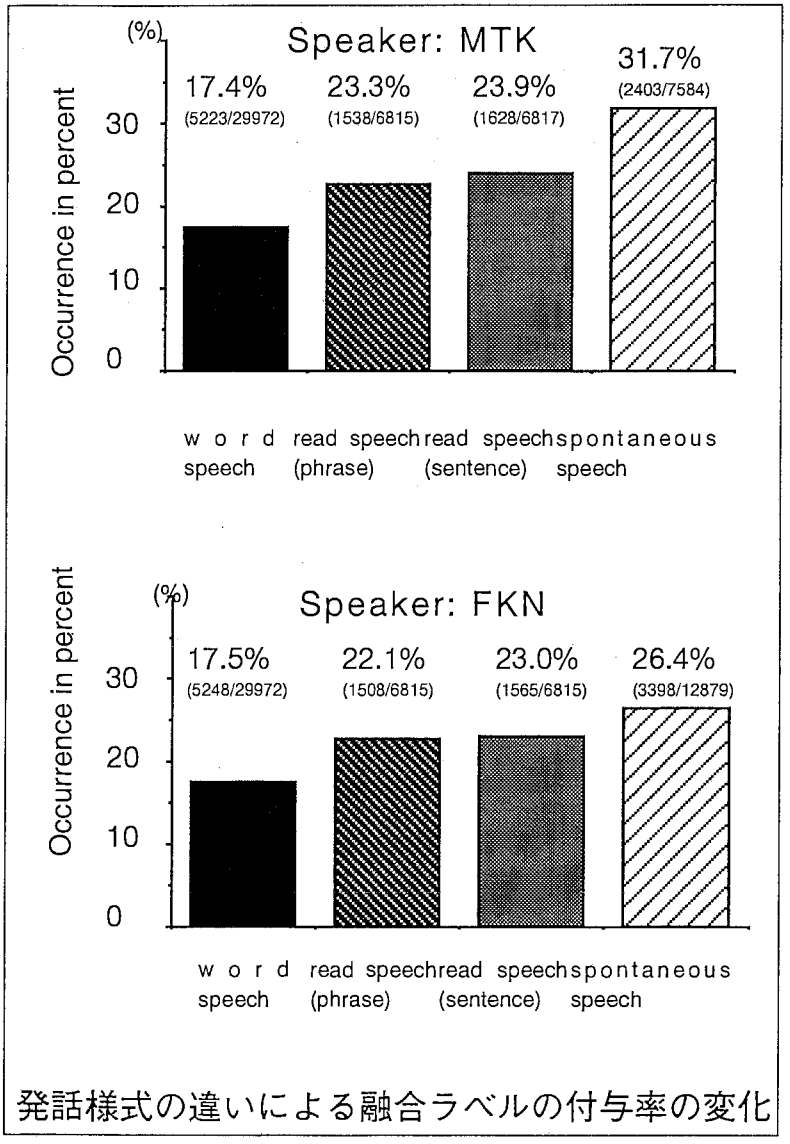


図 3.11: 発話様式の違いによる融合ラベルの付与率の変化

1. 自由発話と文の朗読発声を比較すると、融合ラベルの付与率は話者 MTK では 33%(23.9% → 31.7%)、話者 FKN では 15% (23.0% → 26.4%)、増加する。

2. 音素別に自由発話と文の朗読発声を比較すると、母音では /a/ の増加が顕著である (MTK:4.0% → 13.3%, FKN:3.9% → 8.1%)。子音では、 /m/ の増加が著しい (MTK:1.0% → 18.1%, FKN:1.6% → 10.8%)。

3. 自由発話では、全音素の約 1/4 以上が融合ラベルになる。

4. 単語発声・文節単位の朗読発声・文単位の朗読発声では融合ラベルの付与率に話者の相違は見られない。しかし、自由発話では話者の相違が見られる (MTK:31.7%, FKN:26.4%)。

5. 文節単位の朗読発声と文単位の朗読発声を比較すると、融合ラベルの付与率にあまり差がない。

6. 単語発声・文節単位の朗読発声・文単位の朗読発声・自由発話の順に融合ラベルの付与率が増加する。

3.4.2 発話速度からみた自由発話音声

発話様式における発話速度の違いを調べるために、同一話者における 4 種類の発話様式（単語発声、文節単位の朗読発声、文単位の朗読発声、自由発話）におけるモーラ速度の差を調査した。これを図 3.9に示す。ただし、調査の際、息つぎなどの長いポーズ区間は除去した。また融合ラベルを付与された音素は音素継続時間の計算から除いた。

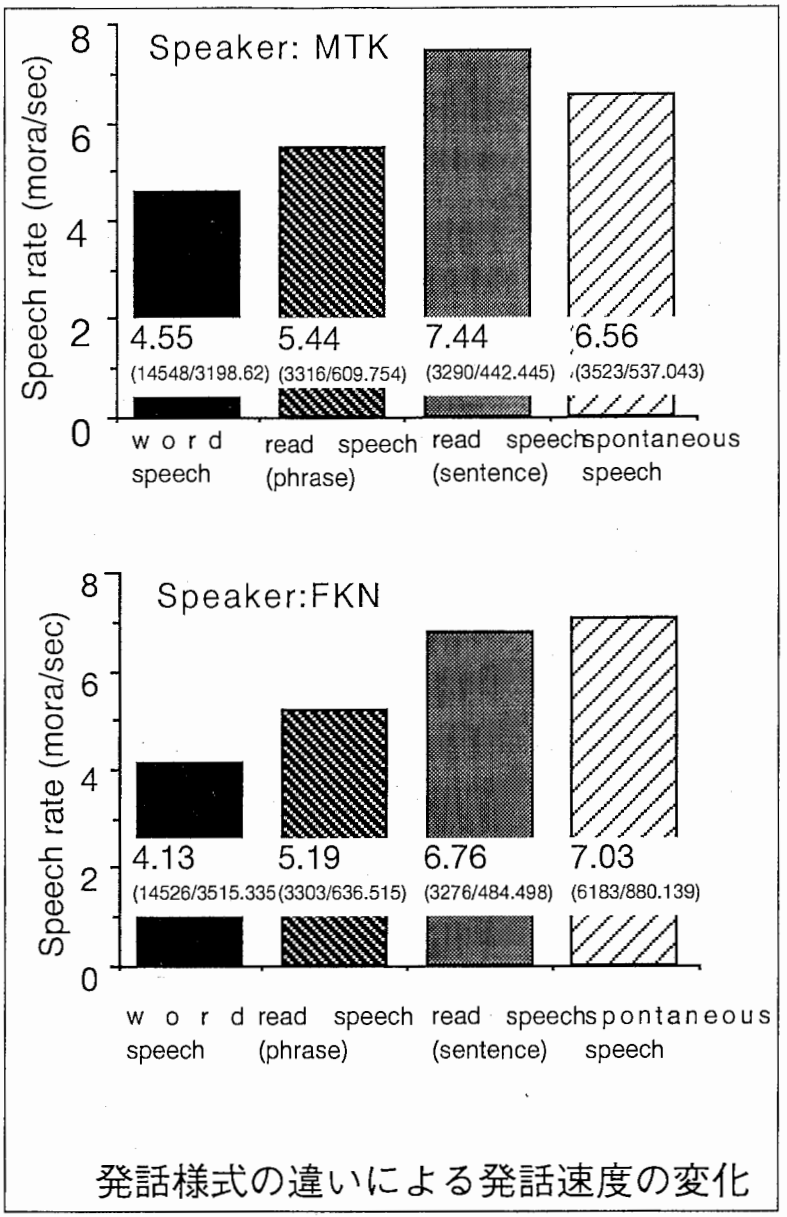


図 3.12: 発話様式の違いによる発話速度の変化

この結果から以下のことが示される。

1. 自由発話の発話速度は、話者 MTK では文単位の朗読発声より早いが、話者 FKN では文単位の朗読発声より遅い。

2. 自由発話における母音の平均音素継続時間は朗読発声より長い。しかし子音の平均音素継続時間は朗読発声より短かい。また、自由発話の音素継続時間の分散は朗読発声より大きい。これは、自由発話では音素の音素継続時間に大きなバラツキがあることを意味する。

3. 発話速度は単語発声・文節単位の朗読発声・文単位の朗読発声の順に早くなる。

3.4.3 音素認識誤り率から見た自由発話

ここでは各発話様式の差を音素認識誤り率で評価した。認識アルゴリズムには混合連続分布型 HMM を用いた。ただし、融合ラベルを付与された音素は実験では用いなかった。また学習

データとして単語発声から視察によって切り出した音素を使用した場合と、同一発話様式の音声データから視察によって切り出した音素を使用した場合の、2種類の実験を行なった。

実験は表3.8とほぼ同一である。たたし、学習データに単語発声を使用した場合、HMMの混合数は10mixtureで、その他は3mixtureである。学習データに単語発声を使用した場合の、各発声様式における音素認識誤り率を、図3.13に示す。また、同一発話様式の音声データを2つにわけ、一方を学習データとし、一方をテストデータとして実験した場合の音素認識誤り率を、図3.14に示す。これから次のような結果が示される。

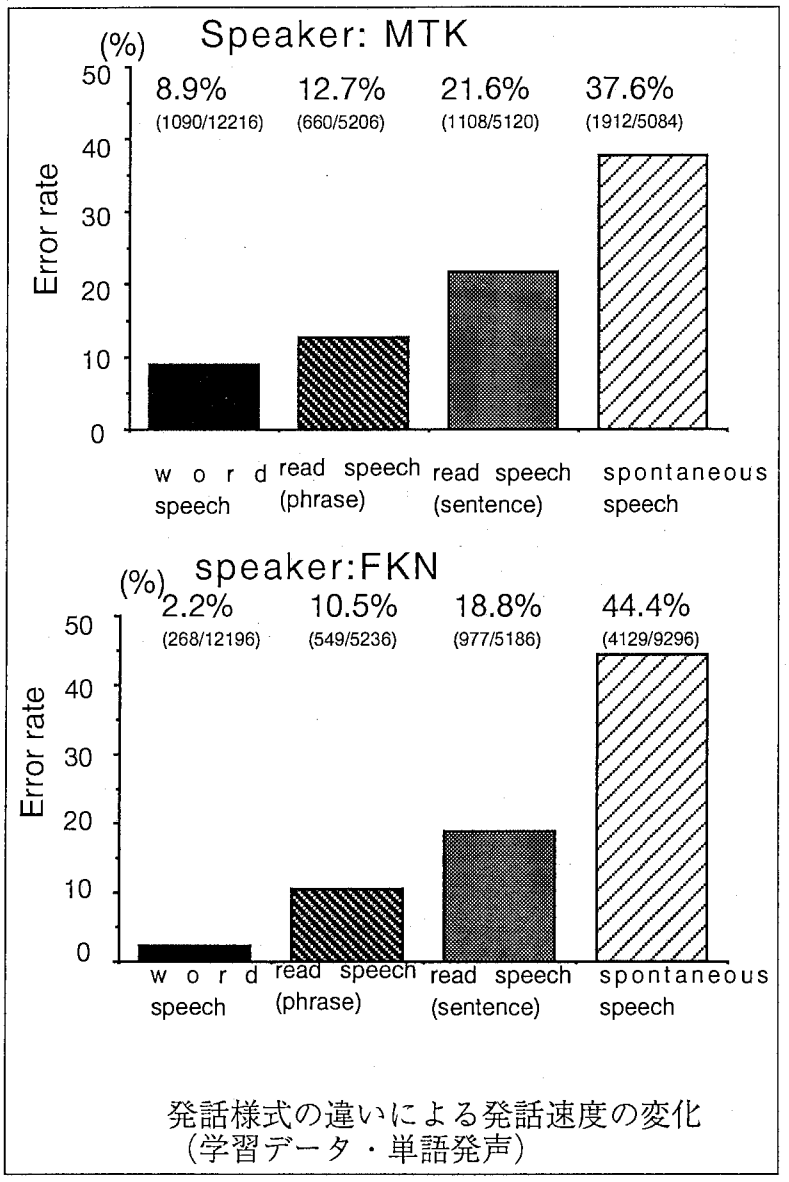


図 3.13: 発話様式の違いによる発話速度の変化（学習データ単語発声）

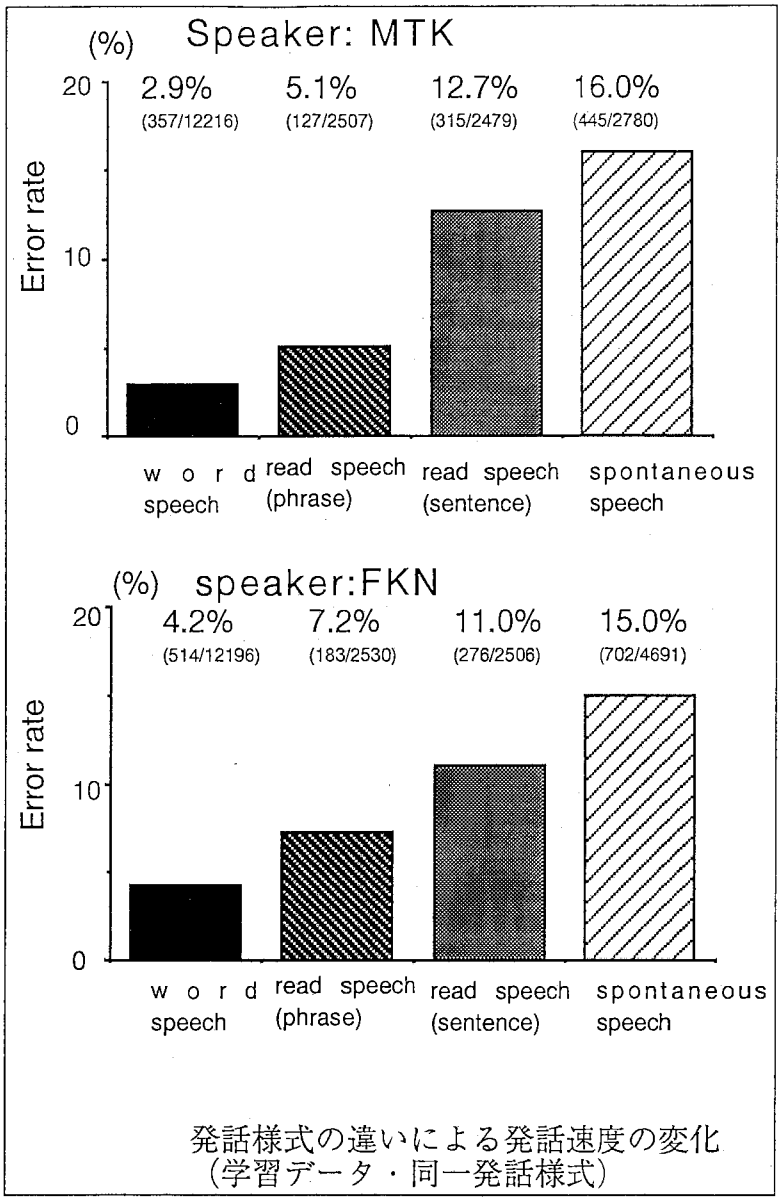


図 3.14: 発話様式の違いによる発話速度の変化（学習データ同一発話様式）

1. 学習データが単語発話のとき、自由発話の音素認識誤り率は高い。朗読発声の音素認識誤り率と比較すると、ナレータ MTK は約 160% 程度増加し (21.6% → 37.6%) ナレータ FKN では約 240% も増加している (18.8% → 44.4%)。

2. 学習データに自由発話の音声を利用することにより、音素認識誤り率は大きく低下する (MTK:37.6% → 16.0%, FKN:44.4% → 15.0%)。学習データが単語発声のときの文の朗読発声の音素認識誤り率 (MTK:21.6%, FKN:18.8%) より低くなる。

3. 自由発話を学習データとした場合、母音の中では /u/ の認識誤り率が高い (MTK:43.9%, FKN:27.9% また、調査音素の数が少ないため明確ではないが、子音では /w/ の認識誤り率が高い。(MTK:78.9% FKN:66.7%)、

4. 単語発声、文節単位の朗読発声、文単位の朗読発声、自由発話の順に音素認識誤り率が増加する。

5. 学習データが同一発話様式の場合、各発話様式において話者の相違はあまり見られないが、学習データが単語発話のとき、話者の相違が見られる。

3.5 考察

3.5.1 間投詞の出現頻度と種類に関して

今回の調査では、話者によって相違があるが、間投詞が出現する文は文章全体の 40% から 65% を占めることが示された。しかし、電話のような音声のみによる対話では、間投詞は相手の注意を促す役割を持つ場合がある [13]。したがって話し相手と対面して話す自由発話では、この出現頻度より低くなる可能性がある。

なお、自由発話における間投詞（冗長語、不要語）の出現頻度は多くの研究期間で報告されている。文献 [25] や文献 [55] では日本音響学会連続音声データベースの書き起こしテキストを調査して報告している。また、文献 [13] では、開始符合としての間投詞の種類と出現頻度を報告している。また、文献 [48] では NHK ラジオ第一放送の電話相談番組を書き起こして報告している。これらの論文と比較すると、間投詞の出現頻度はほぼ同じ割合と言える。また、間投詞の種類も、これらの報告と比較すると比率に違いがあるが、代表的な間投詞に関してはほぼ同じ割合といえる。

3.5.2 自由発話における言い直しに関して

今回の調査では、話者によって相違が見られるが、言い直しを含む文は文章全体の 2% から 4% を占めることが示された。しかし、今回の調査した話者はナレータ（アナウンサや声優など音声を職業としている人）であるため、一般の人の言い直しの出現頻度は、これよりも高いと思われる [36]。

なお、文献 [54] では言い直した単語に着目して、言い直しを分析している。これを見ると言い直しの 59% は、言い誤った単語を直ちに言い直している。また文献 [45] においてもほぼ同様な結果が見られる。今回の自由発話データの言い誤りを分析すると、単語にならない音節となっているものが 39%、直後に言い直しているのが 52% であり、傾向はほぼ同じであった。

3.5.3 自由発話と朗読発話の音響的な差

本論文では、自由発話の音響的な特徴を調査するために、主に融合ラベルの付与率、発話速度、HMM における音素認識誤り率で朗読発話と比較した。その結果、自由発話は朗読発話と比較すると、発話速度は最も差がある話者でも 6% しか増加しないが、融合ラベルの出現頻度は約 20% も増加する話者がいることが示された。しかし、自由発話と朗読発話の認識精度 (Phone accuracy) の差は 7% から 10% 程度であることが示された。また、音素認識誤り率は、単語発声の音声データを学習に使用したとき 2 倍以上になるが、自由発話の音声データを学習に使用した場合は約 3 割増加することが示された。また、学習データが自由発話のときの、自由発話の約 15% という音素認識誤り率は、学習データが単語発話のときの朗読発声の文認識よりも低い。

したがって、少なくとも同一話者（特定話者）、同一発話様式で HMM を学習をする限り、音響モデルに関しては自由発話と朗読発話に大きな差はないように思われる。

ただし、本論文で調査した話者は音声による対話に慣れた人である。したがって、一般の話者が雑音下で制約の少ない状態で話した音声では、この論文で調査した結果と異なる可能性がある。

3.5.4 自由発話の可能性について

自由発話において特徴的な言語現象に、間投詞や言い直し・言い誤り・言い淀みなどがある。そして、今回の調査の結果、間投詞は発話全体の 40% から 65% の文に、言い直しは約 2% から 4% の文に出現することが示された。自由発話の認識には、これらの言語現象の処理方法が大きな問題になると考えられる。

現在自由発話の認識アルゴリズムとしては、これらの現象に対応するため、1) キーワードスポッティングを利用する方法 [59]、2) 音素モデルにガーベージモデルなどを使用して認識する方法 [17][16]、3) 言語モデルの一部に音素系列として認識する方法 [30],[25] もしくはこれらの組合せの手法 [60] などが試みられている。しかし、これらのアルゴリズムには挿入誤りが増加することや、広いビーム幅が要求されるなどの問題点が残っている。

3.6 まとめ

ここでは自由発話の認識にむけて、4 名のナレータにおける自由発話の音響的な特徴を調べた。この結果、話者によって相違が見られるが、間投詞が出現する文は文章全体の 40% から 65% を占めることや、間投詞全体の出現頻度の 50% から 75% は 4 種類の間投詞で占めることがわかった。また、言い直しは文章全体の 2% から 4% に出現することがわかった。

そして、視察による音素ラベリングの結果、自由発話は朗読発話と比較すると、発話速度は、最も差がある話者でも 6% しか増加しないが、融合ラベルの付与率は約 20% も増加する話者がいることがわかった。しかし連続音素認識の実験の結果、認識精度 (Phone Accuracy) は朗読発話と比較すると 6% から 10% しか減少しないことがわかった。したがって、これらの点を考慮すると、間投詞や言い直しなどの言語現象を除けば、少なくとも音素モデルに関しては、同一話者（特定話者）、同一発話様式において音素の HMM の学習をしたならば、自由発話と朗読発話に大きな差はないように思われる。

ただし、ここで扱った自由発話は、言葉の対応に慣れた人たちが限定した条件の下で発話したデータである。したがって、一般の話者が、雑音下で制約の少ない状態で話した音声では、この論文で調査した結果と若干異なる可能性がある。

第 4 章

連続音声認識システム

4.1 連続音声認識のアルゴリズム

連続単語認識アルゴリズムとして最も基本的なアルゴリズムはフルサーチである。この他に2段 DP や One-pass DP、level building などのアルゴリズムが知られている。ここではまず始めにフルサーチと Viterbi サーチ (One-pass DP) のアルゴリズムについて説明する。

4.1.1 フルサーチ

連続単語認識アルゴリズムとして最も基本的なアルゴリズムは、フルサーチである。

このアルゴリズムは、テストデータ全てに対して全ての可能性を計算するため、計算量、メモリ量は膨大になる。しかし、N 位までの累積尤度の単語列 (N-best リスト) を出力することができる。また、グリッドの選択において最尤なものを選ぶ方法 (Viterbi) とグリッドの尤度を足す方法 (Trellis) の両者が選択できる。Trellis で計算をした場合、状態を明確に考慮する必要がないため、Duration control は基本的に必要としない。そして、最終フレームにおいて単語の HMM の最終状態を意味するグリッドの中で、尤度の最も高いものを選択することで文を認識するため、基本的に traceback は必要としない。したがって任意の時間において単語を認識が可能なため、単語スポッタとしても動作が可能である。

またアルゴリズムにおいて各単語の HMM の最後の状態と後続する単語の最初の状態の遷移において任意の言語モデルの制約を加えることにより、音響モデルと言語モデルを簡単に結合することができる。つまり、言語モデルは単語 bigram に限らず CYK などの全ての left-right 型の言語モデルを採り入れることが可能である。

このアルゴリズムを表 4.1 および図 4.1 に示す。

表 4.1: 連続単語認識におけるフル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
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
$G_t(w_n,..,w_0,i)$：単語 w_n から単語 w_0 までの状態 i でのフレーム t までの最大累積尤度

［初期化］

$w_0 = 0,...,Q-1$ において step1 を実行
1) $w_n = 0,...,Q-1$ において step3 を実行
.
.
.
2) $w_0 = 0,...,Q-1$ において step3 を実行
3) $G_0(w_n,...,w_0,0) = 0.0$

［単語内での計算］

$t = 0,1,..,T-1$ において step4,step8 を実行
4) $w_n = 0,...,Q-1$ において step7 を実行
.
.
.
5) $w_0 = 0,...,Q-1$ において step7 を実行
6) $i = 0,1,...,l_{w_0}-1$ において step7 を実行
7) $G_t(w_n,...,w_0,i) =$
$\Sigma(G_{t-1}(w_n,...,w_0,i) \times a_{i,i}^{w_0} \times b_i^{w_0}(O_t),$
$G_{t-1}(w_n,...,w_0,i-1) \times a_{i-1,i}^{w_0} \times b_{i-1}^{w_0}(O_t))$

［単語境界の計算］

8) $w_n = 0,1,...,Q-1$ において step10 を実行
.
.
.
9) $w_0 = 0,1,...,Q-1$ において step10 を実行
10) $\Delta = \Sigma(G_{t-1}(0,w_n-1,...,w_0,l_{w_0}-2)$
$\times a_{l_{w_1}-2,l_{w_0}-1}^{w_0} \times b_{l_{w_0}-1}^{w_0}(O_t),$
$G_t(w_n,...,w_0,0))$

図 4.1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_a と w_b の 2Word 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。図中の
①は時間 0 から時間 $t-1$ までの単語 w_a の状態 0、②は時間 0 から時間 $t-1$ までの単語 w_a

の状態1、③は時間0から時間$t-1$までの単語w_aの状態2、④は時間0から時間tまでの単語w_aの状態2、⑤は時間0から時間$t-1$までの連続2単語w_a, w_aの状態0、⑥は時間0から時間$t-1$までの連続2単語w_a, w_bの状態0、⑦は時間0から時間tまでの連続2単語w_a, w_aの状態0、⑧は時間0から時間tまでの連続2単語w_a, w_bの状態0、の累積尤度であるとする。

フルサーチのtrellis計算においては、単語の最初の状態0を意味するグリッド以外は、前時刻の同一状態を意味するグリッドの尤度と前時刻の1つ前の状態のグリッドの尤度の2状態を加えて現時刻のグリッドの尤度を計算する。例えば、④は①の遷移と②から遷移の累積尤度の総和とする。

しかし、単語の最初の状態0を意味するグリッドは、前時刻の同一のグリッドの累積尤度とレベルが1つ下の単語の最終状態を意味するグリッドの累積尤度の総和とする。例えば、⑦は③の遷移と⑤の遷移の累積尤度の総和とする。

これを全グリッドに対して計算を行なう。

なお、このとき言語モデルの確率値を掛けることにより音響モデルと言語モデルが結合できる。例えば、⑦ は③の遷移と⑤の遷移の累積尤度に単語bigram$P(w_b|w_a)^{\alpha}$, ($alpha$: language weight)を掛けることによって、単語のbigramと単語のHMMが簡潔に結合できる。

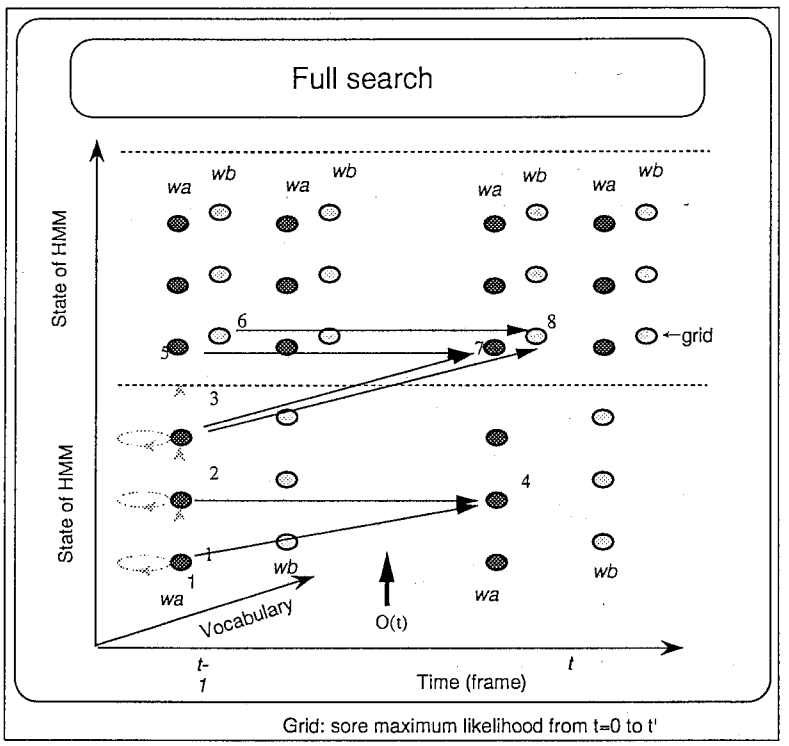


図 4.1: フルサーチのアルゴリズム

4.1.2 Viterbi サーチ (One-pass サーチ)

Viterbi サーチ (one-pass DP) は各認識単語の最後の状態を意味するグリッドと単語の最初の状態を意味するグリッドの遷移において尤度の高い遷移を選択することで計算がされる。認識単位を単語とした場合のアルゴリズムを表4.2に示す。

表 4.2: 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 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 $G_t(w_0, i)$：単語 w_0, 状態 i での フレーム t までの最大累積尤度
[初期化]
$w_0 = 0, ..., Q-1$ において step1 を実行 1) $G_0(w_0, 0) = 0.0$ *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 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) $G_t(w_0, i) =$ $\max(G_{t-1}(w_0, i) \times a_{i,i}^{w_0} \times b_i^{w_0}(O_t),$ $G_{t-1}(w_0, i-1) \times a_{i-1,i}^{w_0} \times b_{i-1}^{w_0}(O_t))$
[単語境界の計算]
7) $w_0 = 0, 1, ..., Q-1$ において step8 を実行 8) $\Delta = \max_{0 \le w_1 \le Q-1}(G_{t-1}(w_1, l_{w_1} - 2)$ $\times a_{l_{w_1}-2, l_{w_1}-1}^{w_1} \times b_{l_{w_1}-1}^{w_1}(O_t) \times P(w_0 \mid w_2, w_1)^\alpha$ もし $\Delta \ge G_t(w_0, 0)$ ならば $G_t(w_0, 0) = \Delta$

図 4.2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_a と w_b の 2Word 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。①は時間 $t-1$ において単語 w_a 状態が 0、②は時間 $t-1$ において単語 w_b 状態が 1、③は時間 t において単語 w_a 状態が 0、④は時間 $t-1$ において単語 w_b 状態が 2、⑤は時間 $t-1$ において単語 w_a 状態が 2、⑥は時間 t において単語 w_b 状態が 2、を意味するグリッド（最大累積尤度）であるとする。

単語の最初の状態を意味するグリッド以外は、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、⑥は②の遷移と④から遷移の最大累積尤度の高い方を選択する。しかし、単語の最初の状態を意味するグリッドは、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の高い方を選択する。例えば、③は①,⑤,(遷移の尤度の高い方を選択する。

なお、単語の bigram を利用するときは、③は⑤に bigram の値 $(p(w_a|w_a)^\alpha)$ を掛けたものと④に trigram の値 $(p(w_a|w_b^\alpha)$, $(alpha$: language weight) 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。

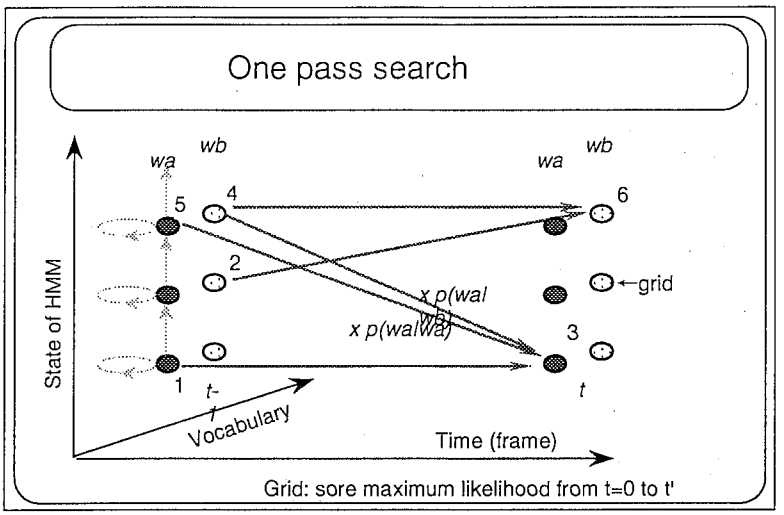


図 4.2: One-pass のアルゴリズム

4.1.2.1 Viterbi サーチにおける N-best サーチ

通常 Viterbi サーチでは、第 1 候補しか出力できない。しかし、最大累積尤度 $G_t(w_0, i)$ を N 個用意することにより、1 回の forward サーチで N-best の単語列が出力できる [1]。図 4.3 に、単語 bigram を使用したときの例を示した。この図では語彙は (A,B,C,D) の 4 単語とし、4-best の場合を示している。

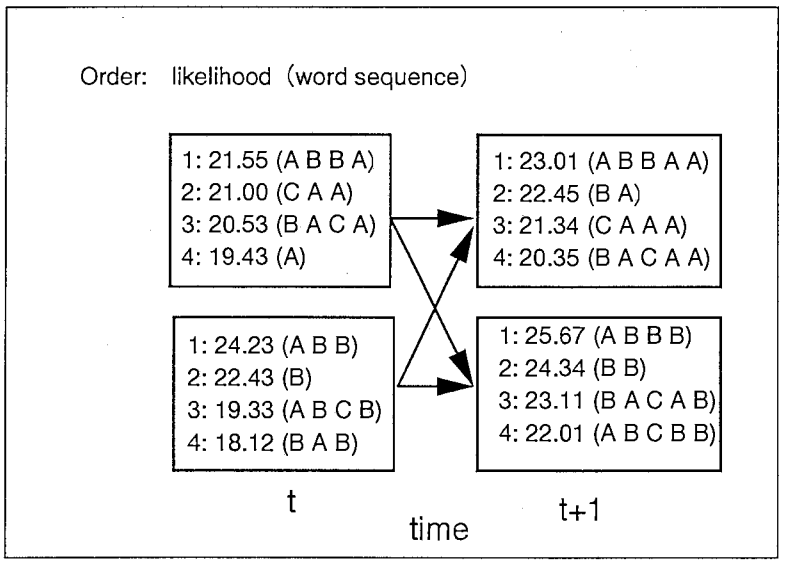


図 4.3: N-best の計算方法
N-best list calculation method

4.1.2.2 Viterbi サーチの経路計算

Viterbi サーチにおいて最尤の単語列の結果を得るアルゴリズムとして、2 つの方法が考えられる。

1. 最大累積尤度の計算終了後にトレースバック

   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 [27]。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 O(認識語彙数 $\times$ 音声データのフレーム数) のメモリ量が必要である。

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。図 4.4) に単語 bigram を使用した場合の例を示す。単語 trigram を使用したときもほぼ同様なアルゴリズムになる。このアルゴリズムにおいて必要なメモリ量は O (認識語彙数 $\times$ 文の単語数) である。ただし、この方法は、経路をコピーする必要があるため計算量は前の方法と比較すると、若干増加する。

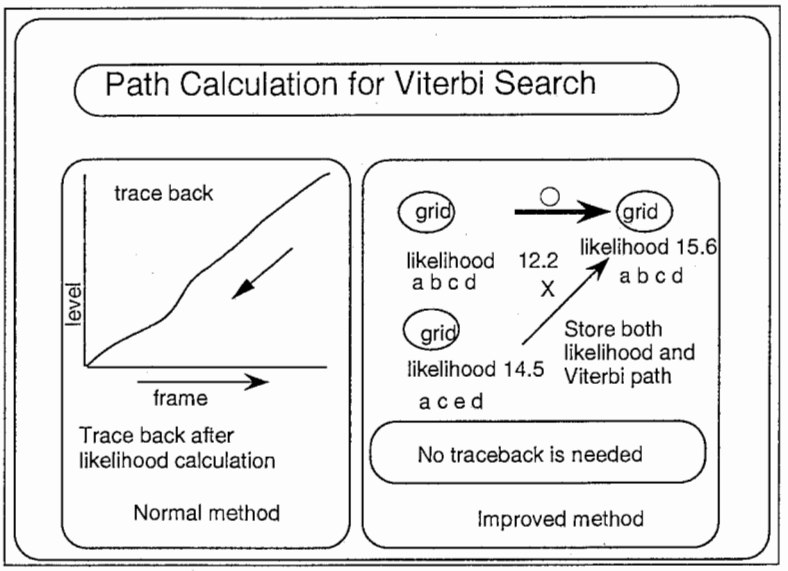


図 4.4: Viterbi サーチの経路計算方法
Viterbi path calculation method

前者は、計算量が少なくて済むため広く利用されている。後者は、前者と比較すると計算量は若干増加するが、多くの場合、文の単語数は音声データのフレーム数より少ないためメモリ量が削減できる。なお、このアルゴリズムは各時刻・各状態 ($G_t(w_0, i)$) においてトレースバックをしなくても累積尤度が最大の単語列を知ることができる。

4.1.2.3 単語の trigram を使用したときの Viterbi サーチ

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 4.3に示す。

表 4.3: 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 $G_t(w_1,w_0,0)=\Delta$

図 4.5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 0、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 1、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 1、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 2 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 2、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 0、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 0 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。

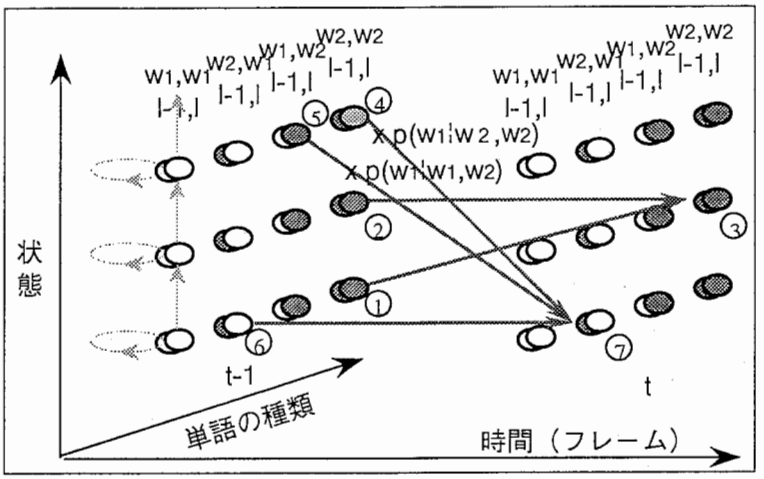


図 4.5: 単語の trigram を利用したときの Viterbi サーチ

4.1.3 フルサーチと One-pass サーチの比較

上記で、フルサーチと One-pass のアルゴリズムを述べた。この両者のアルゴリズムにはそれぞれ特徴がある。認識性能と計算コストとメモリー量を考慮してアルゴリズムを選択する必要がある。表 4.4にこれらのアルゴリズムの特徴についてまとめる。

表 4.4: フルサーチと One Pass サーチの比較

	フルサーチ	One-pass サーチ
計算コスト	大きい	小さい
メモリ	大きい	小さい
グリッドの選択方法	Viterbi & trellis	Viteribi
N-best list	可能	アルゴリズムを改良して可能
言語モデルとの適合性	Left-Right 型の全ての言語モデルが可能	Left-Right 型の全ての言語モデルが可能。ただし近似解になる。
ビームサーチとの適合性	良好	良好
音素モデルにおける duration control との適合性	良好（ただし Trellis 計算では必要としない。）	良好
スポッタとしての動作	可能	プログラムを改良すれば可能

4.1.4 オブジェクト グリッド

上記で、フルサーチと One-pass のアルゴリズムを述べた。しかし、グリッドを中心に考えると、フルサーチにおいて 1 単語ごとにマージするものを、One-pass サーチと呼んでいることになる。

また、音素の HMM は前後の音素環境を考慮しない context independent タイプと同様に context dependent タイプも使用されている。

グリッドを考えるとき、前後の音素環境も考慮しながらマージすることにより triphone のような context-dependent model も扱える。また、言語モデルとしてネットワーク文法や文脈依存文法などを利用するとき、過去の履歴に関して完全に一致する場合のみマージをすることで、left-right 型のネットワーク文法に当てはめることが可能である。

これはグリッドを中心に考えることにより、統一がとれる。

4.2 計算量およびメモリ量の削減方法

フルサーチでは大量のメモリと計算量が必要になる。そこでメモリ量と計算量を削減するために、アルゴリズムを次に述べるように改良する。

4.2.1 ビームサーチ

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正解の単語列になる可能性が低いため、以後の探索から除外できる可能性が高い。そこで、フレームごとに最も高い累積尤度から正解の存在をおおよそ保証できる、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削減できる [49]。具体的には、すべての $w_n,..,w_0,i$ に対して表 4.1 、10) 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 $G_t(w_1,...,w_0,i)$ を記憶するメモリ量は、フルサーチでは O(認識語彙数^n× 単語の状態数) が必要であるのに対し、ビームサーチでは O(ビーム幅 b) しか必要としないため大幅に削減できる。また、計算量もビーム幅の計算方法によって異なるが、同様な比率で削減できる。

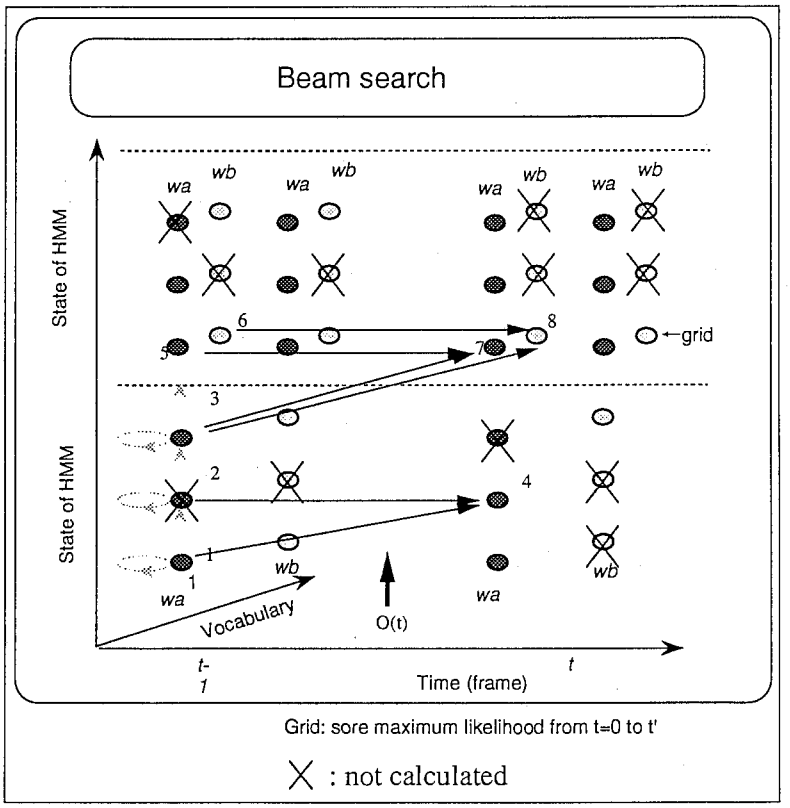


図 4.6: ビームサーチの計算方法

4.2.2 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

   尤度の閾値でビームを絞る方法は、計算量が少なくてすむためよく利用されている [27]。しかし、認識を行なう前に予め閾値を決めておかなければならないため、動作が不安定になることがある。

2. ビーム幅

   一定のビーム幅でビームを絞る方法は、フレームごとにソーティングが必要になる。そのため、計算量が増大する。

本論文では、後者のアルゴリズムを選択している。だたし、計算順序として、始めにフレームごとに最も高い最大累積尤度 (G_{max}) からビーム幅 b の最大累積尤度 (G_b) を計算し、次にこの尤度 (G_b) でビームを絞り込んでいる [70]。この方法はフルソートと比較すると計算量が大幅に削減できる。

4.2.3 近接したフレームにおける言語モデルの類似性の利用

音声認識アルゴリズムにおいて単語と単語の境界の尤度計算するとき、言語モデルを使用したばあい、パーザーを動かして単語仮説を生成させる必要がある。しかし、フレーム同期型のアルゴリズムでは、直前のフレームにおいて生成された文仮説候補は、現在のフレームの文仮説候補になる可能性が高い。この特徴を利用して、前回の単語仮説の確率値を利用することで再度言語モデルをパーズする必要がないため、大幅に計算量が削減できる。なお、この方法は文献 [19] においても紹介されている。

4.2.4 単語 trigram の値の記憶

言語モデルとして単語の trigram を利用する場合、単語 trigram の値を直接記憶すると［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 [71] を採用することにより、trigram の値を参照するための計算量を削減できる。

4.2.5 log 計算

音声認識において連続分布の HMM を使用したとき、計算のダイナミックレンジが大きく変化するため、対数をとって計算をおこなう。そのとき対数同士の足し算が必要になる。このアルゴリズムとして、文献 [3] において線形補間の方法が報告されている。ここでは別の方法を採用した。 $A = \log(a)$

$B = \log(b)$

$C = \log(a + b)$

$= \log(a) + \log(1 + b/a)$

$= A + \log(1 + \exp(B - A))$

を利用して以下のようにした。

$if(A \gg B) \quad A;$

$$
\begin{aligned}
&else\ \ if(B \gg A) \quad B;\\
&else\ \ if(A \geq B) \quad A + \log(1 + \exp(B - A));\\
&else\ \ if(B \geq A) \quad B + \log(1 + \exp(A - B));
\end{aligned}
$$

4.2.6 音素 HMM

表 4.3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。しかし、単語の HMM は音素の HMM を連結させて作成するのが一般的である。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。

4.2.7 Look ahead 処理

通常の言語モデルでは言語の確率を計算してから音響の尤度を計算する。この順序を逆にする。つまり言語の制約をあとから付け加える。これにより、単語が認識されてから、言語モデルが駆動される形にある。そのため、実質的にビームが少なくて済む。ただし、この方法は認識率を低下させるため、実験では用いていない。

第 5 章

trigram の有効性について

文節単位の音節マトリックス形式で入力された日本語音節認識候補から漢字かな混じりの日本語文節候補を生成する処理において、従来の音節の二重マルコフモデルによる音節の文節候補の選択に加えて、漢字かなの二重マルコフモデルを適用して漢字かな混じりの文節候補の絞り込み効果を２つの方法で、実験的に明らかにした。　第１の方法（音節選出型文節処理方式）では、初めに文節単位の音節マトリックスに音節の二重マルコフモデルを適用して音節の文節候補を得る。次に単語辞書を参照して単語候補に変換する。最後に、漢字かなの二重マルコフモデルを適用して、漢字かな混じりの文節候補を生成する。　第２の方法（直接選出型文節処理方式）では、初めに同音節マトリックスから同単語辞書を用いて直接、単語候補を抽出する。次に、漢字かなの二重マルコフモデルを適用して漢字かな混じりの文節候補を生成する。　それぞれの方法における正解率を求めた結果、文節候補の生成において、漢字かなの二重マルコフモデルの効果は顕著で、第１位の候補の正解率は第１、第２の方法でオープンデータにおいて、それぞれ 65%、70%、クローズドデータにおいて、それぞれ 83,84% となり、高い精度の漢字かな混じりの文節候補が得られることが分かった。

5.1　はじめに

日本文音声入力においては、音声の持つ物理的特性に着目した音声認識装置の限界を克服するため、日本語の文法や意味を用いた自然言語処理を併用することの必要性が指摘されている [62]。特に大量語彙を対象とする音声には発音の個人差や曖昧さの他に、同音異義語なども多数含まれるため、その認識においては音声の物理的特性が完全に生かされたとしても、なお絞り切れない曖昧さが残り、元の文を推定するには、言語解析や意味理解の技術が必要と考えられる。音響処理と自然言語処理を融合させた、日本文音声入力の一つの方法として、文節単位の音節マトリックスをインターフェースに用いて、音声認識装置と自然言語処理を連携させる方法 [29] が考えられている。すなわち、音声認識装置が音声の物理的特性を解析して、文節単位に各音節候補をマトリックス形式で出力し、自然言語処理はそのマトリックスを入力として、正しい漢字かな混じりの文節候補を推定する方法である。この場合の言語処理の方法としては、従来、二つの方法が考えられる。その一つは、音節マトリックスに言語の文法情報や意味情報を直接適用して、正しい文節を推定しようとするもの [57] であり、もう一つは、音節や文字の統計的な連鎖情報を適用して文節候補を絞り込む方法][51] である。　前者は文法、意味情報を直接適用して文節を生成する点に特徴がある。しかし、実際の単語の代わりに単語の文法的カテゴリーや意味的カテゴリーが使用されるため、絞り込みの精度はこれらのカテゴリーの分解能に依存し、複数の単語候補が同一のカテゴリーに属するような大量語彙の認識では、文節候補を絞り込むのは困難である [57]。一方、後者の方法では、大量語彙の認識において、音節の二重マルコフモデルが有効で、その適用により、文節単位の音節マトリックスから、第一位で約 70%、第１０位までの累積

正解率で約95%の高い精度の音節文節候補を生成できることが指摘されている[2]。しかし、漢字かなの文節候補を生成するにはさらに膨大な曖昧性を絞り込むことが必要であった。　ところで、漢字かな混じりの文の誤字、脱字等に漢字かなのマルコフモデルが効果的であること[15]が知られている。そこで本論文では、音節マトリックスから文節候補を生成するための方法として、音節の二重マルコフモデルのほかに漢字かなの二重マルコフモデルおよび単語辞書を使用した。そして、これらを組み合わせた二種類の曖昧性絞り込みの方法を提案し、その効果を実験的に示した。

5.2　実験システムの構成

5.2.1　日本文音声認識の処理手順

日本文の音節や文字連鎖の持つ情報量を応用した「文節処理」の効果を調べるため、音声の持つ物理的特性に着目した音声認識処理と、それ以外の言語処理的な部分とを分け、図5.1に示すような日本文音声認識手順を考える。　日本文の音声入力のマンマシーンインターフェースとしては、単音節単位、文節単位および連続音声の入力などが考えられるが、ここでは音声認識装置は一音節単位に認識した複数の音節候補を文節の単位で出力、つまり文節単位の音節マトリックスの形態で出力するものとする。「文節処理」では、音声認識装置から出力された音節マトリックスから単語候補を生成し、その中で適切と見られる漢字かな混じりの文節候補を、その数を限定して出力する。最後に「文処理」において文節候補を文単位に結合して得られる最も適切な文節の組を入力文に対する認識結果として出力する。　以下では、以上の日本文認識手順の中の「文節処理」において、日本文の音節や漢字かなの二重マルコフモデルを用いた認識候補絞り込みの方法を提案し、その効果を実験的に示す。

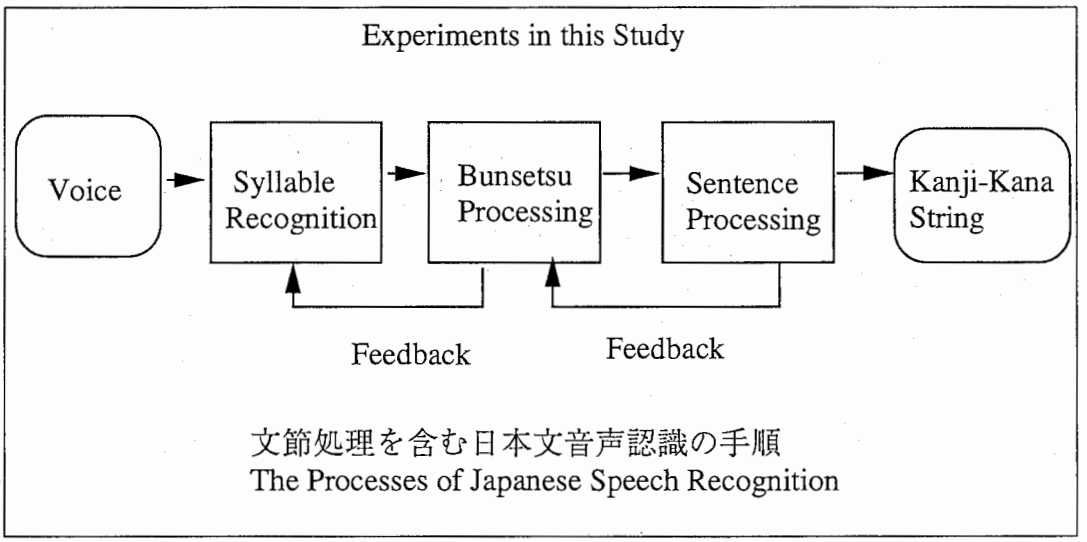


図 5.1: 仮想的な音声認識システム

5.2.2　文節処理の方法

「音声認識処理」と「文節処理」は図5.2に示すような文節単位の音節マトリックスで結合されるものとし、「文節処理」の結果としては、文節毎の漢字かな混じり文を出力する。

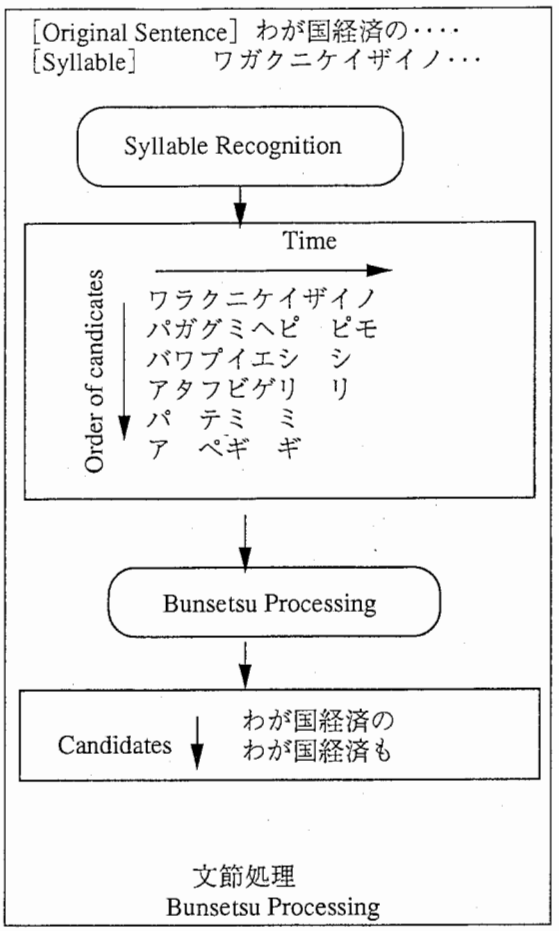


図 5.2: 文節処理の出力例

大量語彙を対象とする漢字かな混じり文の生成では、同音異義語が多数存在し、同一のかな列に対して複数の漢字が対応するため、音節列の場合 [2] に比べて曖昧さが桁違いに大きく、通常、数億個以上の候補が出力される。従って、「文節処理」の課題は、このような膨大な文節候補の中から、正解を含む少数の文節候補を選択することである。 以下では、このような「文節処理」の方法として図 5.3に示す二つの方法を考える。

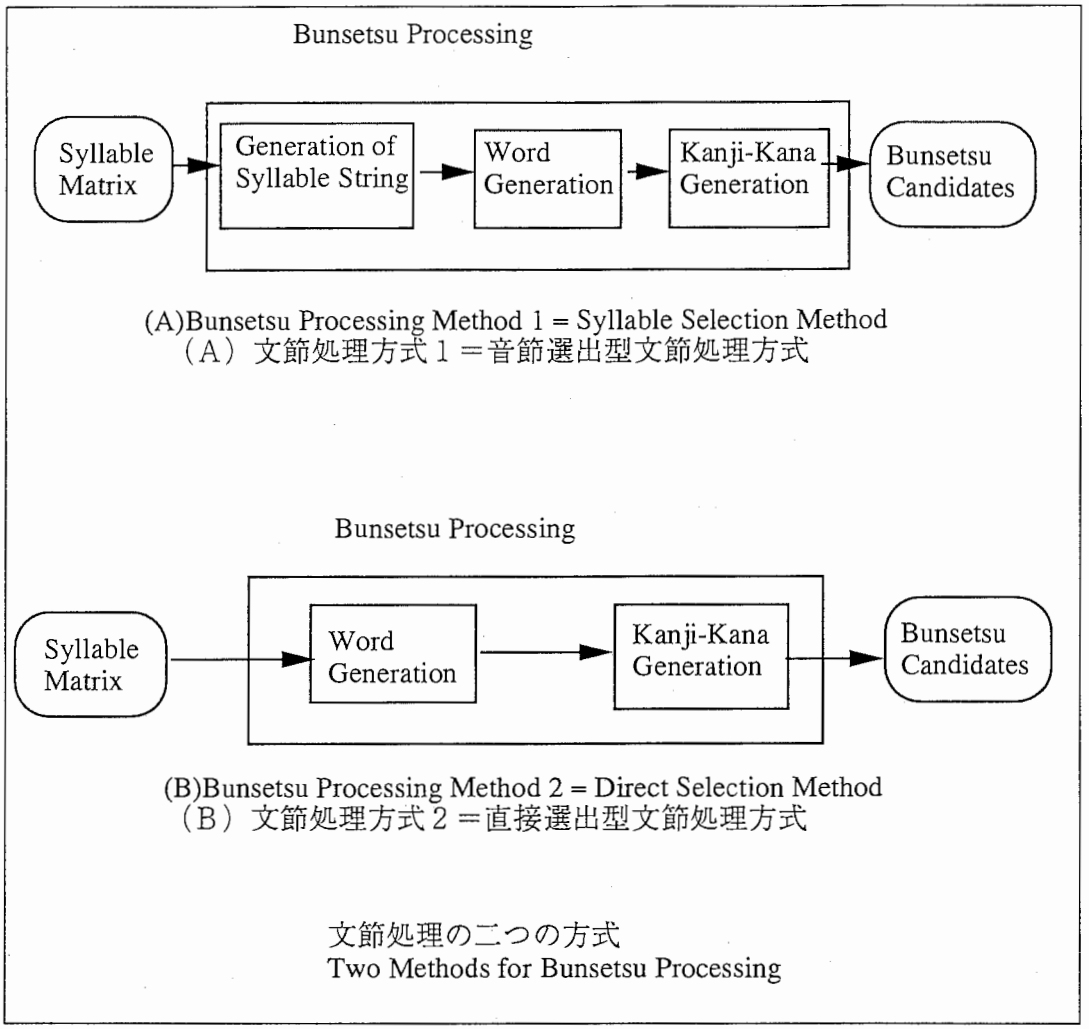


図 5.3: 文節処理の 2 つの方式

1. 音節選出型文節処理方式

入力された文節単位の音節マトリックスから以下の 3 ステップの処理を経て、文節候補を生成する。すなわち、まず初めに、音節マトリックスに対して日本語の持つ音節のマルコフモデルを適用して音節の組み合わせ候補を絞り込む。次に、その結果に対して単語辞書を適用して文節を構成する単語候補を生成する。最後に、漢字かなのマルコフモデルを使用して、文節を出力する。

2. 直接選出型文節処理方式

上記の方法が、はじめに音節連鎖情報を使用するのに対して、この方法は、音節マトリックスに直接単語辞書を適用するもので、以下の二ステップで文節候補を生成する。　はじめに単語認定においては音節マトリックス内の音節候補を組み合わせながら辞書引きを行い、単語として解釈可能な候補の組み合わせをすべて抽出する。次に漢字かなのマルコフモデルを使用して文節候補を生成する。

5.3 文節候補生成アルゴリズム

5.3.1 音節選出型文節処理のアルゴリズム

音節選出型文節処理方式における入出力データの流れを図5.4に示す。

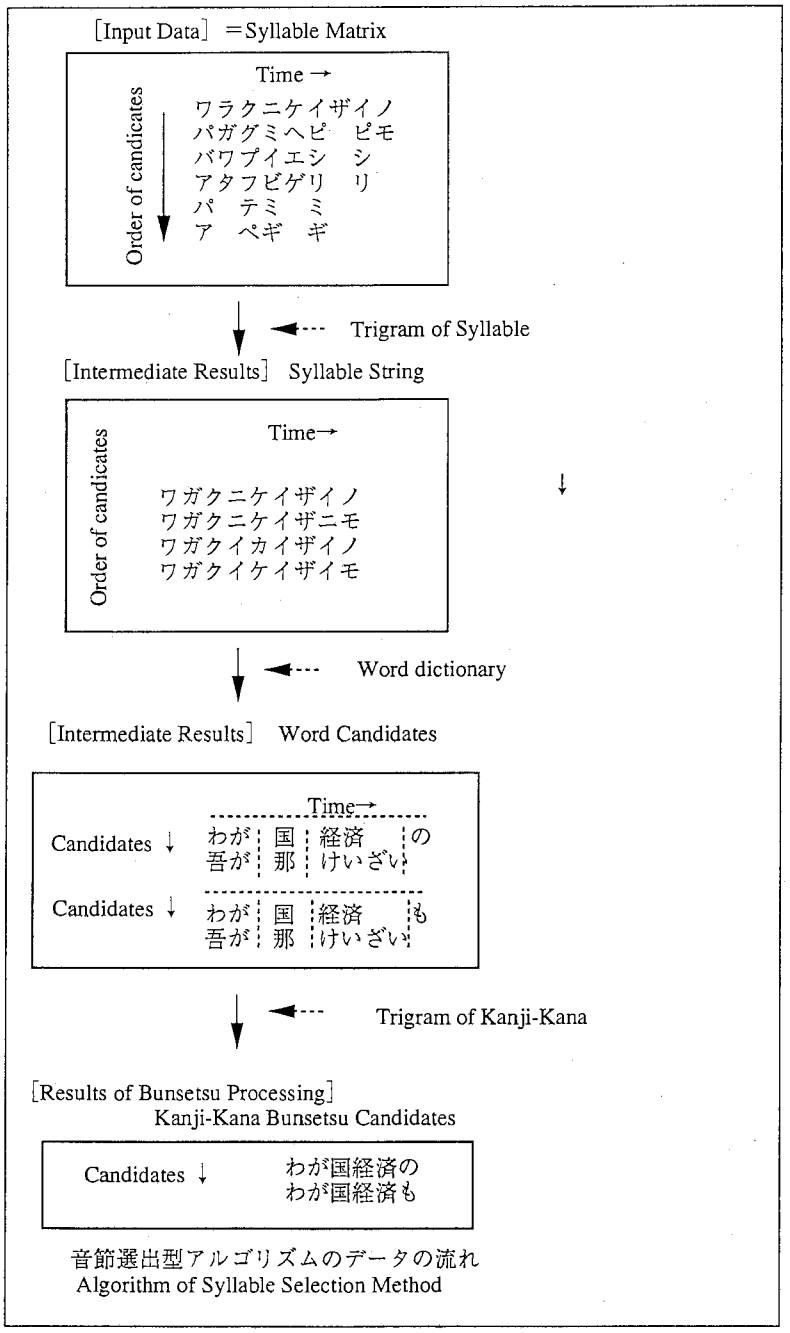


図 5.4: 音節選出型文節処理方式における入出力データ

1. 音節文節候補の生成アルゴリズム

音節マトリックスから音節文節候補を生成する方法として、二重マルコフモデルを用いる。マルコフモデルによる候補絞り込みは、正しい文節候補は間違った候補よりもマルコ

フ連鎖値の積が大きいと仮定して、文節候補を評価する。例えば図5.4の例で、「ワラクニケイザイノ」の文節候補の尤度は

$p($ _ _ ワラクニケイザイノ_ _ $)$ $= p($ ワ／_ _ $) \times p($ ラ／_ ワ $) \times p($ ク／ワラ $) \times p($ ニ／ラク $) \times p($ ケ／クニ $) \times p($ イ／ニケ $) \times p($ ザ／ケイ $) \times p($ イ／イザ $) \times p($ ノ／ザイ $) \times p($ _ ／イノ $) \times p($ _ ／ノ $)$.

（ただし _ は空白を意味する。）で与えられる。これを他の音節の組み合わせを含む１６５，８８８通りのすべてについて計算し、上位何候補かに絞り込む。この場合は第１位の候補として「ワガクニケイザイノ」が得られ、第２位としては「ワガクニケイザイニモ」が得られる。　一般に、音節マトリックスを対象に直接この計算を行うのは計算量の点で困難であるが、ビテルビのアルゴリズムを使用することにより、少ない計算量で容易に評価することができる。

2. 単語認定アルゴリズム

前項で得られた複数の音節列の上位８位までの音節列に対して、単語辞書を参照し、当てはまる単語候補を出力する。　このプロセスはワードプロセッサのかな漢字変換と基本的に同じである。ここでは分割数最小法 [42] を基本とするが、正解候補のもれを防止するするため、最小分割数＋１までの単語候補を生成する。

3. 文節候補認定アルゴリズム

最後に上記で得られた単語候補に対して漢字かなの二重マルコフモデルを使用して曖昧性を絞り込む。なお実験では同時に品詞の二重マルコフモデルを使用して、品詞における文節候補の絞り込みの効果も調べた。

5.3.2　直接選出型文節処理のアルゴリズム

直接選出型文節処理方式における入出力データの流れを図5.5に示す。

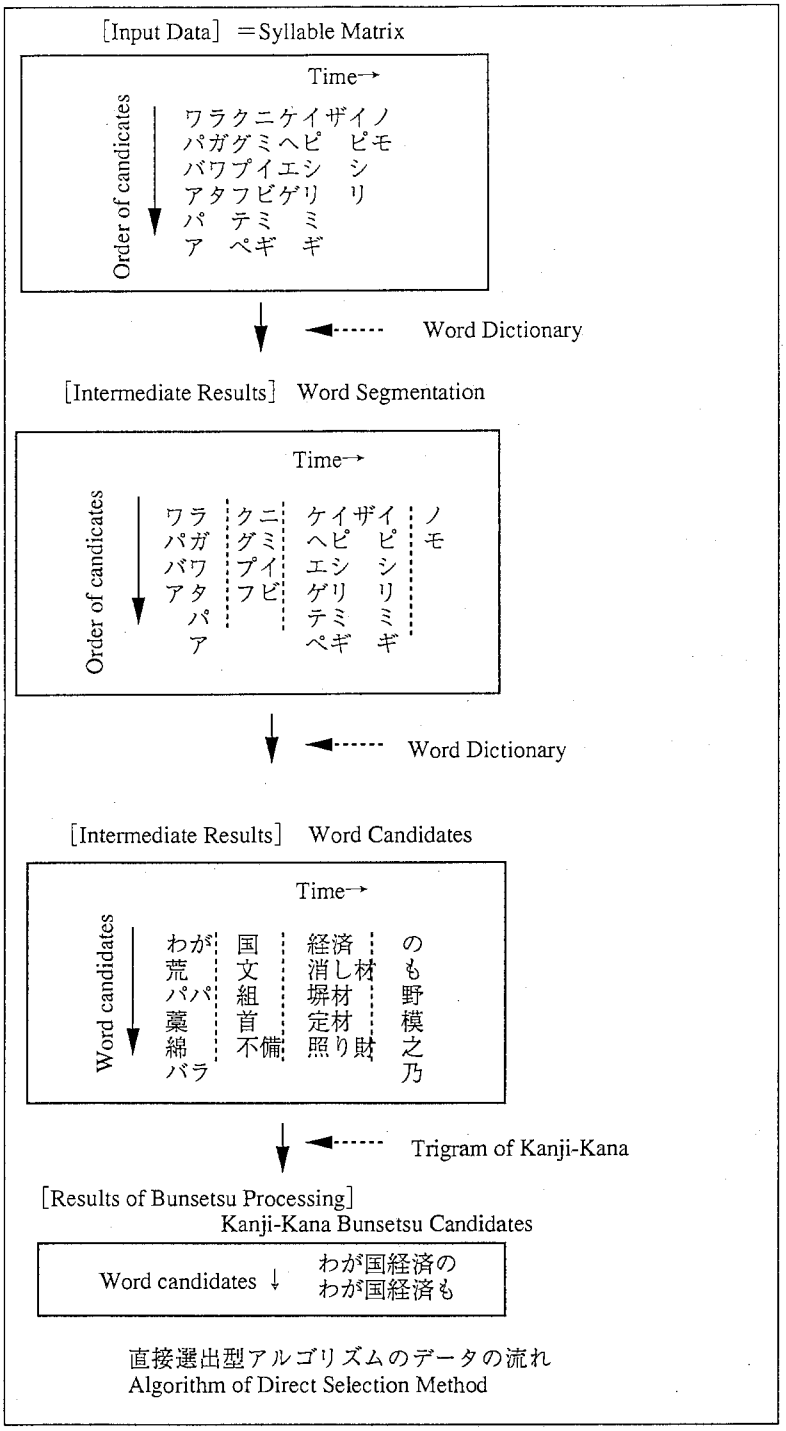


図 5.5: 直接選出型文節処理方式における入出力データ

1. 単語認定アルゴリズム

文節単位の音節マトリックスに以下の方法で直接単語辞書を適用し、可能な単語候補をすべて抽出する。 まず音節マトリックスの音節候補をつなぎ合わせた音節列の中に文節を一単語として解釈できる単語候補があるかどうかを単語辞書を使って調べる。図5.5の例では、9音節を一単語と考え、各音節を組み合わせた単語の有無を調べる。すなわち、$4 \times 6 \times 4 \times 4 \times 6 \times 6 \times 1 \times 6 \times 2 = 27648$ 通りの音節の組み合わせに対して、9音節の全てが一致するような単語が辞書に存在するか否かを調べ、存在すればすべて抽出する。もしそ

のような単語が存在しなければ音節マトリックスを二つに分割する。図5.6の例ではそのような単語候補はないので、下記（実線）のようにマトリックスを二つに分割する。

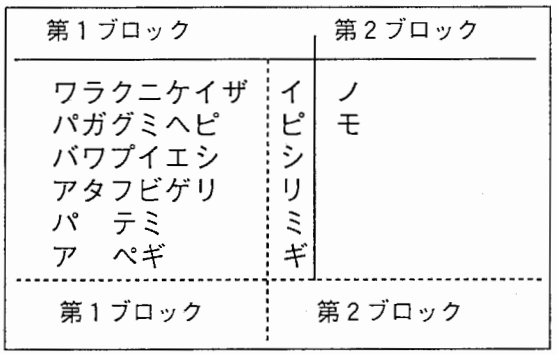


図 5.6: 入力データの一例

第１ブロック、第２ブロックの双方に対して前と同様の方法で単語辞書引きを行い、辞書上の単語の有無を調べる。何れかのブロックに対して単語が存在しないときは分割が不適切と考え、第１、第２のブロックの分割の仕方を変える（破線）。　分割された二つのブロックの双方に一つ以上の単語候補が存在するような分割の仕方が無いときは、全体を三つのブロックに分割する。全てのブロックに対して一つ以上の単語候補が存在するようになるまで、この手順を繰り返し、辞書上で解釈可能な最小の分割数を求める。　また、このようにして求めた分割数最小の分割法の全てに対して、ブロック毎に辞書上解釈可能な全ての単語候補を出力する。　図５は最小分割数が４で、４ブロックに分割したときの各ブロックに対する単語候補を示している。

2. 文節候補認定アルゴリズム

前項で抽出された単語候補を組み合わせて得られる漢字かな混じりの文節単語列に対して、同様の漢字かなの二重マルコフモデルを適用し、順位付けを行う。なお実験では同時に音節および品詞の二重マルコフモデルを用いて、それぞれの情報の効果を調べた。

5.3.3　両アルゴリズムの違いについて

音節選出型文節処理方式と直接選出型文節処理方式のアルゴリズムでは、使用される情報は同じであるが、その適用順序に違いがある。前者は音節の二重マルコフモデルを最初に使用するので、その後、評価対象となる候補数が大幅に減少する。そのため、全体としての計算量が少ないと言う利点があるが、逆に単語辞書の適用の段階で、正しい文節候補が失われている可能性がある。　これに対して、後者のアルゴリズムでは単語辞書を最初に適用するため、多数の単語候補が生成され、後の処理が重くなるが、正しい漢字かな混じり文の文節候補をもたらす可能性は、より高いと予想される。

5.4　実験方法

5.4.1　実験の条件

1. マルコフ連鎖値

マルコフ連鎖値の計算には日経新聞記事７４日分（８２年１月４日から３月３１日）を使用した。これを日本文解析プログラムを使用して形態素に分割し、同時に音節変換を行った。そして、これを再合成して文節単位のデータを作成し、その後、音節、漢字かな、品詞について０重、一重、二重のマルコフ連鎖値を計算した。

ただし実験を簡単にするため、この記事から、記号、外国語読み、数詞の文字のある文は文全体を削除した。その結果、マルコフ連鎖値の計算に使用した文字数は漢字かな混じり文字で数えて約１７０万文字である。

なお、新聞記事は、マルコフモデルに必要な、すべての組み合わせを持っていない。そのため、連鎖値が０となる組合せが出現する。そのような組み合わせに対しては、統計上の最小値を与える方法や次数の少ない連鎖値との補間で代用する方法 [21] などが考えられるが、ここではフロアリングをして確率値を exp(-1000.0) とした。

2. 音節マトリックス

文節処理の入力となる音節マトリックスは、従来の音声認識装置 [9] の認識率情報（コンフュージョン・マトリックス）に基づき、以下の条件でコンピュータ・シミュレーションにより生成した。

(a) セグメンテーション誤りはないものと仮定する。

(b) 音節候補の数は最大８個とし、８位までの候補の中に必ず正しい音節候補があるものとする。

(c) 音節の認識距離情報は使用しない。すなわち、音節マトリックスにおける候補順位は無視し、全て同一の重みと仮定する。

(d) 音節に長音「ー」、鼻音「カ゜」行、促音「ッ」の存在を仮定する。これは音声出力用の形式で登　　録されて単語辞書とのインターフェースを合わせるためである。なお、これらの音節の１位正解率は 100% としている。

また、実験には以下の２種類の音節マトリックスを用意した。

(a) オープンデータ

マルコフ連鎖値の計算に使用した日本文以外の漢字かな混じり文から生成した文節単位の音節マトリックス。（日経新聞８２年１月１日の記事文から抽出）

(b) クローズドデータ

マルコフ連鎖値の計算に使用した日本文の漢字かな混じり文から生成した文節単位の音節マトリックス（日経新聞８２年１月５日の記事文から抽出）

3. 単語辞書

単語辞書は一般語、使用頻度の高い人名地名などの固有名詞を含む１６万語の日本文音声変換用の辞書を使用した。ただし、使用した情報は音節、漢字かな、品詞の三種類である。

5.5 結果と考察

5.5.1 実験結果

直接選出型文節処理方式において失敗した文節例と成功した文節例をそれぞれ図5.7、図5.8に示す。実験の結果得られた音節、漢字かな、および品詞の文節候補の右端に示した数値は二重マルコフモデルの総積値の自然対数の逆数を文字数で割った値である。したがって値が小さいほど尤度が高いことを示している。

[Correct Data]

Syllable	ハンカク	シュウカイ	ハ
Kanji-Kana	反核	集会	は
Part of Speech	一般名詞	サ変名詞	副助詞

[Input Data] =Syllable Matrix

ハ	ン	カ	ク	シュ	ー	ガ	イ	ワ
タ		タ	プ	チュ		カ	ピ	ア
カ		パ	フ	ヌ		ア	リ	バ
ア		チャ	グ	ツ		タ	シ	パ
パ		ア		チャ		ミ		
		ガ		ハ		ギ		
		ハ				パ		

[Final Results]

（1） Syllable

Order	Syllable	Value
1	カンガクシューカイワ	2．24
2	ハンパクツーカイワ	2．29
3	ハンタクシューカイワ	2．31
4	カンカクシューカイワ	2．35
5	ハンパクシューカイワ	2．35
6	ハンガクシューカイワ	2．39
7	ハンカクシューカイワ	2．34
8	カンタクシューカイワ	2．40

（2） Kanji-Kana

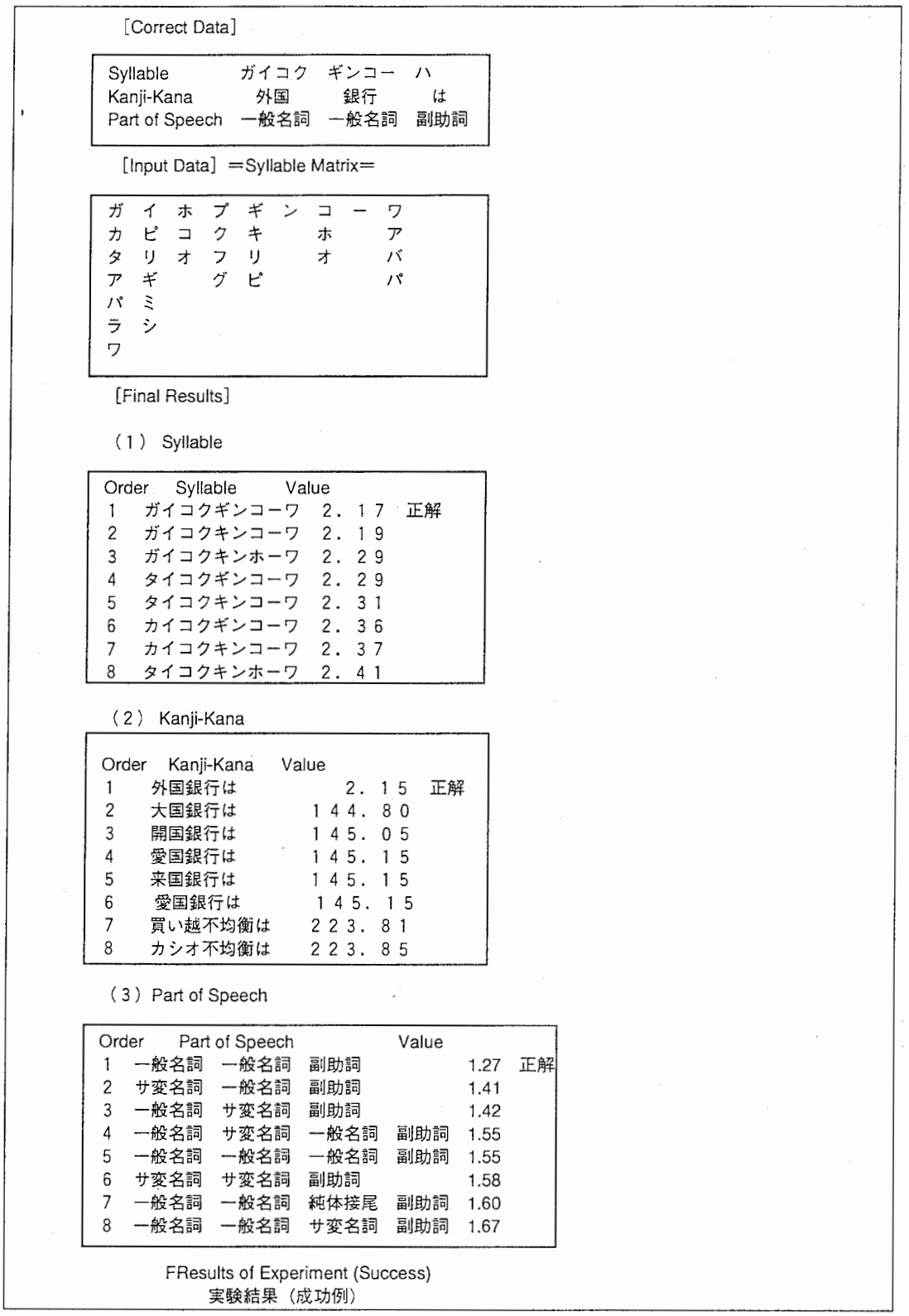
Order	Kanji-Kana	Value
1	たんぱくちゅうたいわ	169．17
2	タンパクちゅうたいわ	169．33
3	たん白ちゅうたいわ	184．68
4	たんぱく通貨市場	202．26
5	タンパク通貨市場	202．45
6	感覚ちゅうたいわ	202．66
7	反核ちゅうたいわ	202．72
8	間隔ちゅうたいわ	202．85

（3） Part of Speech

Order	Part of Speech				Value
1	一般名詞	一般名詞	副助詞		1.27
2	サ変名詞	一般名詞	副助詞		1.41
3	一般名詞	サ変名詞	副助詞		1.42
4	一般名詞	サ変名詞	純体接尾	副助詞	1.51
5	一般名詞	サ変名詞	一般名詞	副助詞	1.55
6	一般名詞	一般名詞	一般名詞	副助詞	1.55
7	サ変名詞	サ変名詞	副助詞		1.58
8	一般名詞	一般名詞	純体接尾	副助詞	1.62

Example of Experiment (Failure)
実験結果（失敗例）

図 5.7: 直接選出型文節処理方式における誤りの例



[Correct Data]

Syllable	ガイコク	ギンコー	ハ
Kanji-Kana	外国	銀行	は
Part of Speech	一般名詞	一般名詞	副助詞

[Input Data] =Syllable Matrix=

ガ	イ	ホ	プ	ギ	ン	コ	ー	ワ
カ	ピ	コ	ク	キ		ホ		ア
タ	リ	オ	フ	リ		オ		バ
ア	ギ		グ	ピ				パ
パ	ミ							
ラ	シ							
ワ								

[Final Results]

（1） Syllable

Order	Syllable	Value	
1	ガイコクギンコーワ	2．17	正解
2	ガイコクキンコーワ	2．19	
3	ガイコクキンホーワ	2．29	
4	タイコクギンコーワ	2．29	
5	タイコクキンコーワ	2．31	
6	カイコクギンコーワ	2．36	
7	カイコクキンコーワ	2．37	
8	タイコクキンホーワ	2．41	

（2） Kanji-Kana

Order	Kanji-Kana	Value	
1	外国銀行は	2．15	正解
2	大国銀行は	144．80	
3	開国銀行は	145．05	
4	愛国銀行は	145．15	
5	来国銀行は	145．15	
6	愛国銀行は	145．15	
7	買い越不均衡は	223．81	
8	カシオ不均衡は	223．85	

（3） Part of Speech

Order	Part of Speech				Value	
1	一般名詞	一般名詞	副助詞		1.27	正解
2	サ変名詞	一般名詞	副助詞		1.41	
3	一般名詞	サ変名詞	副助詞		1.42	
4	一般名詞	サ変名詞	一般名詞	副助詞	1.55	
5	一般名詞	一般名詞	一般名詞	副助詞	1.55	
6	サ変名詞	サ変名詞	副助詞		1.58	
7	一般名詞	一般名詞	純体接尾	副助詞	1.60	
8	一般名詞	一般名詞	サ変名詞	副助詞	1.67	

FResults of Experiment (Success)
実験結果（成功例）

図 5.8: 直接選出型文節処理方式における正解例

このような出力結果を入力文節１００件について集計した結果を図 5.9と図 5.10に示す。

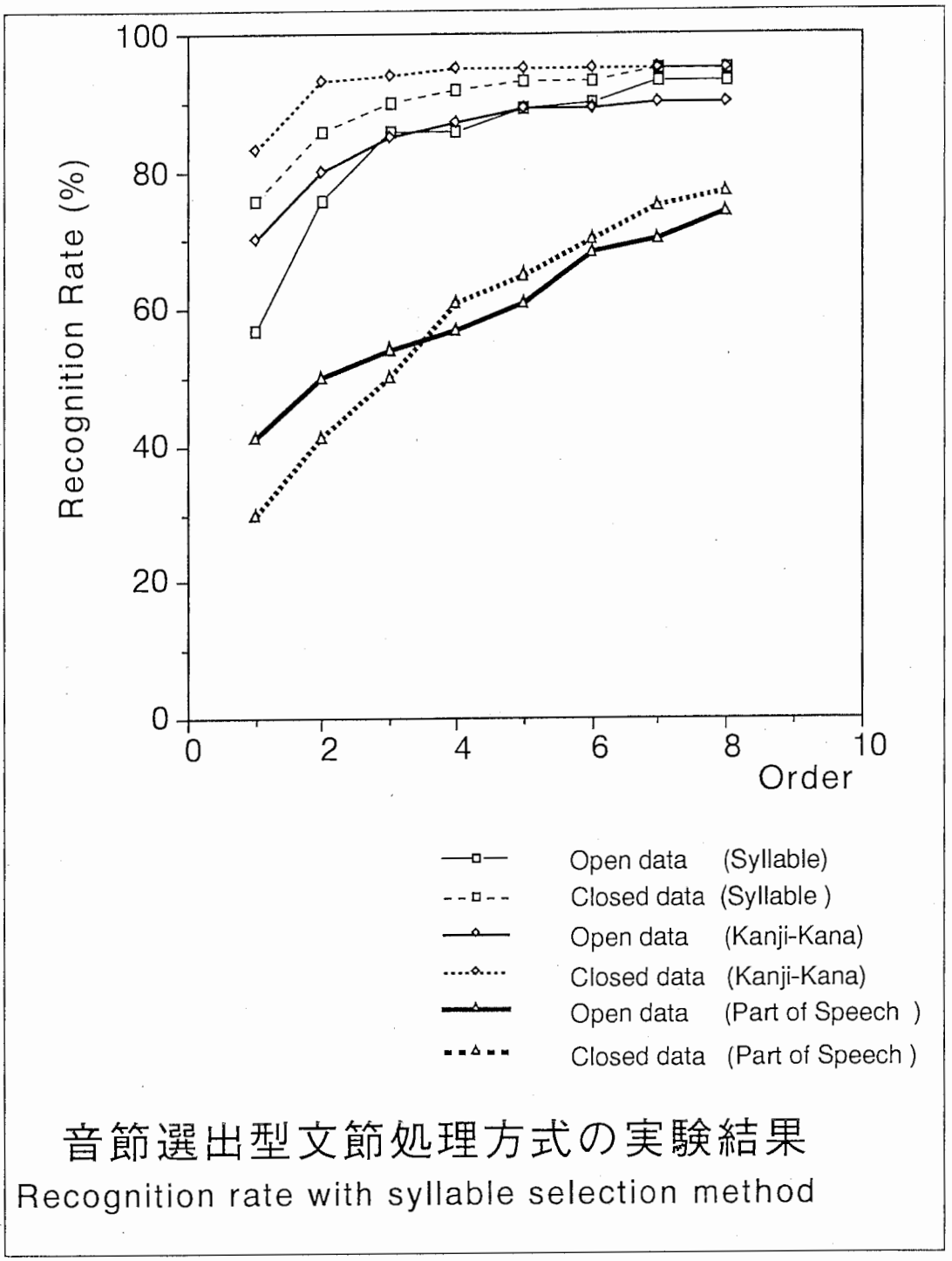


図 5.9: 音節選出型文節処理方式の実験結果

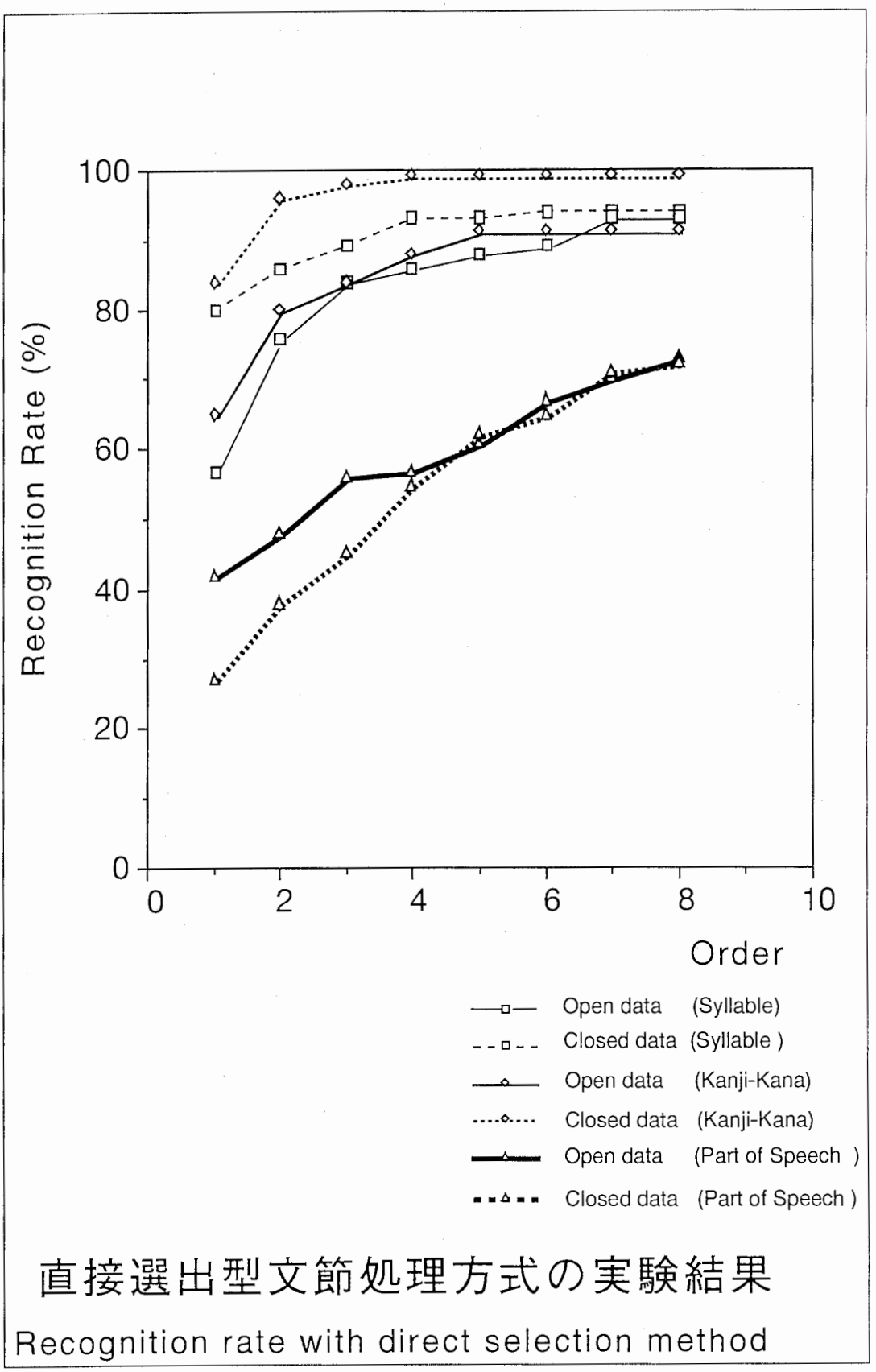


図 5.10: 直接選出型文節処理方式の実験結果

これらの図から以下のことがわかる。

1. 音節の文節候補の1位正解率は最大値が直接選出型のクローズドデータで79%、最小値がオープンデータで56%であった。また、8位までの累積正解率は音節選出型と直接選出型での差はなく、クローズドデータで94%、オープンデータで93%であった。

2. 漢字かなの文節候補の1位正解率は音節選出型でクローズデータでは83%、直接選出型では84%であった。特に直接選出型では4位までの累積正解率は99%を示した。また、オープンデータでは1位正解率が音節選出型は70%、直接選出型では65%であるが、8位までの正解率は共に90%を越えた。

3. 品詞の文節候補の場合は音節や漢字かなの文節候補の場合より正解率が遥かに低く、両方式で見て、1位正解率は27-42%、8位までの累積正解率は72-77%にしか過ぎない。しかし、オープンデータとクローズドデータの正解率の差は殆どない。

5.5.2 考察

1. マルコフ連鎖値の収束性

クローズドデータとオープンデータの正解率の差は音節、特に漢字かな文字の文節候補において顕著であるのにたいして、品詞の文節候補では差がほとんど認められない。これはマルコフ連鎖値の収束性の問題で、さらに多く日本文を収集することにより両者がお互いに接近する形で、その差は減少すると判断される。 なお、マルコフ連鎖値の収束性を調べたところ、品詞、音節、漢字かなの順に、エントロピーの収束性が良いことがわかる。

2. 音節と漢字かなの情報量

音節と漢字かなの特性を比較すると、0，1，3重の場合は音節の方がエントロピーが小さいが、2重の場合は逆に漢字かなの方が小さくなっている点が特徴的である。これは、二重マルコフモデルにおいては、漢字かなの方が情報量が大きく、それ以上、次数を上げても効果は少ないのに対して、音節ではさらに次数を上げればそれだけ効果が得られることを意味していると思われる。

3. 誤りの原因

漢字かなの文節候補の選出において、クローズデータの実験で、正解候補が最終的に8位以内に入らなかった文節を見ると、それらのすべてが、音節選出型の方式では音節の文節候補の失敗に起因し、直接選出型の方式では単語境界の分割数が足りないことに起因していることがわかった。 前者の漏れを防ぐには、音節の二重マルコフモデルで抽出する文節候補の数を増やすことが考えられるが、計算量の増加を伴うので適当なトレードオフが必要となる。また、後者の漏れを防ぐには単なる分割数最小法ではなく、係り受け併用型の分割数最小法[32]を採用した方が良いと考えられる。

5.6 まとめ

日本文音声認識において音声の物理的特性を使用した音声認識装置と自然言語処理の間を結ぶ処理として、二重マルコフモデルを用いた文節処理の二つの方法（音節選出型と直接選出型）を提案し、その効果を実験的に求めた。

その結果、両方の方式とも、漢字かな混じりの文節候補を従来の音節の二重マルコフモデルを用いた文節候補で得られた正解率と同じか、それ以上の精度で、生成できることが分かった。これは、漢字かなの二重マルコフモデルの効果は非常に効果的で、大量語彙辞書を用いて、音節

から漢字かな混じり文を生成する際に生じる膨大な曖昧性がほぼ完全に解消することを意味している。

音節選出型と直接選出型の文節処理を比べると、音節の文節候補の第1位正解率は、後者の精度が若干高いが両者に大きな差異は認められないことから、音節の二重マルコフモデルには単語内の音節のマルコフの情報がかなり反映されており、音節における文節候補の推定の能力の点でみれば、音節間の二重マルコフモデルは単語辞書に代わり得る情報を持つことが推定される。また漢字かなの文節候補では直接選出型の方が精度は高い。これは漢字かなの二重マルコフモデルはかなり大きな情報量を持っているため、単語の候補が増加しても、これが文節候補の推定に影響を与えていないことがわかる。

オープンデータとクローズドデータの場合の比較では両者の差は音節、漢字かなに比べて品詞の場合、差がない。このことから、品詞の二重マルコフモデルは前2者に比べて小量のデータで収束することが分かるが、これは同時に候補絞り込みに使用される情報量が少ないことも意味しており、実験では文節絞り込みの精度は最も小さくなっている。

本論文では、大量語彙の音声認識におけるマルコフモデルの効果を見る立場から、音声認識装置からの認識距離は使用せず、音節、漢字かな、品詞それぞれの2重マルコフモデルの効果について調べた。したがって今後、これらの情報をくみわせた場合について検討する必要がある。

また、本論文では対象外としたが、今後、音声認識部における脱落、挿入などを含むセグメンテーションの誤りの問題や、文節候補の曖昧性をさらに絞り込むための、文節間文法情報や意味、文脈等情報等の適用方法の検討、また、クローズデータにおいて連鎖値が0である場合の値の定め方等の検討が必要である。

第 6 章

単語の HMM と bigram を利用した文節音声認識

ここでは、X 線 CT の所見作成入力用の音声ワードプロセッサーを目指して、認識単位として単語、言語情報として単語の bigram を使用した文節音声認識システムを作成した。語彙数は約 3000 である。このシステムの概要と実験結果について述べる。

6.1 認識単位を単語とした文節音声認識

6.1.1 音響モデル

従来の多くの文（文節）音声認識システムでは認識単位として音節や音素を選択している [41],[27]。しかし、現実の音声データでは音素境界が曖昧な音素が多い。したがって、高い認識性能を目指す場合、長い認識単位が有利であると考えられる。したがって、ここでは認識単位として単語を選択した。しかし、単語を認識単位とした場合、単語の HMM の学習の時に、大量の単語発声の音声データが必要であること、また認識のときに、HMM のパラメータの記憶のために多くのメモリー空間が必要であることなどから、従来はあまり多く行なわれてきていない [31]。そこで、本論文では学習データを減らすため、1 つの単語の HMM の学習に 1 つの単語発声の音声データのみ使用することにした。つまり X 線 CT の所見作成入力用の音声ワードプロセッサーを使用する人に、事前に 3000 単語を 1 回発声してもらい、このデータで単語の HMM を学習した。そして、少ない音声データで精度の高い HMM のパラメータを推定するために Fuzzy-VQ HMM を用いた。また認識時において HMM のパラメータの記憶のためのメモリー空間を減らすために、単語の HMM のモデルは全て 4 状態 3 ループとした。

6.1.2 言語モデル

言語モデルには単語の bigram のみをもちいた。bigram の連鎖確率値の計算には、今まで入手できた X 線 CT の所見作成の全文章、71198 単語から計算した。また、連鎖確率値が 0.0 である場合は $\exp(-1000.0)$ に置き換えた。ただし、deleted-interpolation[6] などの平滑化はおこなっていない。

6.1.3 単語の bigram を用いた文節音声認識アルゴリズム

実験に用いた認識アルゴリズムの基本は、単語の HMM に Viterbi サーチ (one-pass DP) に単語の bigram とした。また実験では HMM の状態 $G(l, w, i)$ を複数 (N 個) 持たせることによって複数の候補を出力する N-best サーチを行なった。

6.2 実験条件

認識実験では duration control と N-best のサーチ幅を変化させて行なった。また、単語の HMM の学習のデータを増加させた場合の実験も行なった。これらの実験の条件を表 6.1に示す。その他の実験条件は表 6.2にまとめた。なお duration control は同一話者の単語発声の 3 回分のデータの平均発声時間と分散を測定し、この値からガウス分布を計算し、duration control に使用した。

表 6.1: 実験条件

実験番号	duration control	N-best	学習データの個数
実験 1	なし	2	1
実験 2	あり	2	1
実験 3	あり	8	1
実験 4	あり	2	3

6.2.1 テストデータ

X 線 CT 所見作成の文章は大きくわけて正常所見と異常所見に分類される。そして異常所見は正常所見と比較すると文章が複雑なため、認識率が低くなることが知られている [57]。そこで実験は、bigram の連鎖確率を計算するのに使用したテキストを発声した音声データ (text-closed data) と bigram の連鎖確率を計算するのに使用しなかったテキストを発声した音声データ (text-open data) について、各々異常所見と正常所見について合計 4 つの条件で行なった。実験は平均 100 文節行なった。

表 6.2: 文節音声認識の実験条件

使用アルゴリズム	word HMM + Viterbi search + word bigram 特定話者認識
認識単位	word
語彙数	約 3000
学習データ	単語発声
言語情報	単語 bigram
音響パラメータ	log power + 16 次 LPCcepstrum + Δlog power
距離尺度	簡易マハラノビス
VQ コード数	256
単語モデル	4-state 3-loop Fuzzy-VQ HMM
フレーム窓長	18ms
フレーム周期	9ms
ファジネス	1.5
近傍数	5
サンプリング周波数	12kHz
HMM と bigram の結合値	32

表 6.3: テストデータの実験

1. text-closed data の正常所見
2. text-closed data の異常所見
3. text-open data の正常所見
4. text-open data の異常所見

6.2.2 実験結果

実験結果を表 6.4に示す。この結果からわかることを以下に示す。

1. 実験 1 から test-closed の正常所見で 96.8%、異常所見では 78.1%、text-open の正常所見でも 86.5%、異常所見では 72.1% の高い文節認識率が得られた。したがって HMM の学習データが 1 つでも Fuzzy-VQ を使用することにより高い文節認識性能が得られることがわかった。

2. 実験 1 と実験 2 の比較から、duration control を行なうと認識性能が低下した。この原因として duration control に使用した平均・分散の値の不正確さが考えられる。これらの値は同一話者が発声した 3 つの単語発声の音声データから計算したため値の信頼度はかなり低い。

3. 実験結果2と実験結果3の比較から、N-bestの幅を広げた方が高い認識率を出すことが示された。

4. 実験結果2と実験結果4の比較から、音声データを増加させることによって認識性能が向上することが示された。これはHMMのパラメータを推定するための学習データが1つでは、不十分であることを示している。しかし不特定話者認識の場合、一人の発話データが1つしかなくても、複数の話者が発話することによって、多くの音声データが利用できるため、認識単位が単語でも問題はないと思われる。

表 6.4: 実験結果

実験番号	1	2	3	4
duration control	なし	あり	あり	あり
N-best	2	2	8	2
学習データ数	1	1	1	3
text-closed data の正常所見	96.8%	82.6%	100.0%	100.0%
text-closed data の異常所見	78.1%	76.3%	78.9%	84.2%
text-open data の正常所見	86.5%	86.5%	89.2%	94.6%
text-open data の異常所見	72.1%	68.9%	72.1%	77.0%

6.3 考察

6.3.1 HMMの種類について

この実験では、HMMの学習に使用する音声データを1つとしたためFuzzy-VQ HMMを使用した。しかし不特定話者認識のためにはcontinuous mixture HMMのほうが相応しいと考えられる。しかし、continuous mixture HMMを使用した場合、mixture数にも依存するが学習に大量の音声データが必要である。そこで、単語を認識単位とするばあいは、学習データがある程度少なくてすむsemi-continuous HMM[3]が有望ではないかと考えている。

6.3.2 認識単位・単語

認識単位として音素を選択したとき、HMMの学習のために、音素ラベルが付与された音声データが必要になる。ラベリング作業は自動化がある程度可能であるが、最終的には人手に頼らざるを得ないため、音声データベースの作成のコストはかなり高い。一方認識単位を単語にしたばあい、ラベリング作業は不用になる。そのかわり、数個の単語発声が必要があるため、発話者の負荷が大きくなる。したがって認識システムの仕様や目的にも依存するが、認識単位を単語としたときのほうが、音声データベースの作成に必要なコストは低くなる可能性があると考えている。

6.3.3 リアルタイムにむけて

音声認識のリアルタイム化には2つの方法がある。1つにはアルゴリズムによる計算量の削減であり、もう1つはハードウエアによる計算コストの分散化である。フレーム同期型の認識アルゴリズムにおいて計算量を削減する方法としてビームサーチが知られている [49]。しかし、超並列コンピュータなどを考えた場合、ビームサーチを採用しないほうが早くなる可能性がある。今後、リアルタイム化はハードウエアも考慮して最適なアルゴリズムを考えていく必要があると思われる。

6.4 まとめ

本報告では、学習データ量に対するマルコフモデルの収束率について調査した。この結果マルコフモデルの連鎖確率の信頼性を調べるためにはエントロピーだけでなく、頻度別出現率も調査する必要があると思われる。また、特定話者の文節認識実験を行なった結果、認識単位を単語とした場合、HMM の学習用の音声データが1つでも、かなり高い認識率が得られること、そして単語の bigram の情報と組み合わせることにより、text-open の正常所見でも 86.5%、異常所見では 72.1% の文節認識率が得られることが示された。

第 7 章

フルサーチと単語の trigram モデルを用いた文音声認識

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 は一般的には bigram 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、大量のメモリ量と計算量が必要になる。本論文では、4.2章で述べたアルゴリズムを基本に朗読発話において単語の trigram を利用したときの認識実験結果について報告する。

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これに対応しきれないため、ポーズを含む音声データは誤認識が起きやすい。ここで利用したフルサーチでは、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の有効性について述べる。

7.1 単語の trigram モデルを用いた文音声認識実験

7.1.1 認識アルゴリズム

ここの実験では、認識アルゴリズムとしてフルサーチを用い、trellis でグリッドを選択した。またフレーム毎にビームサーチをかけている。また、音素の HMM を連結させて単語の HMM を作成した。言語モデルとしては単語の trigram を使用している。

7.1.2 実験条件

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。単語の HMM は音素の HMM を連結して作成した。また音素の 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 2620 単語発声を使用し、不特定話者認識の場合は評価話者とは別の男性話者 12 名の 736 単語発声を利用した。単語の perplexity は trigram で 4.0、bigram で 13.9 である。テストデータは、国際会議の問い合わせのタスクの 261 文で、話者はナレータ 1 名である。実験条件を表 7.1 にまとめる。なお、テストデータの先頭と最後には約 200ms のポーズ区間がある。また、trigram の連鎖確率値は、ATR の対話データベース [7] 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。したがって認識実験は text-closed である。ただしテキストデータ中の「あのー」、「えーと」などの間投詞は削除している。

表 7.1: 文音声認識の実験条件

音素モデル	Continuous mixture HMM
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 2,620 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	1,567
ビーム幅	4,096
継続時間制御	なし
実験文数	261 文, 話者 1 名
発声様式	朗読発話
発声内容	国際会議の問い合わせ
単語 trigram の値の	約 1 万 2 千文章 171,978 単語
推定に使用した	テストデータのテキストを含む
テキストデータ量	（間投詞は削除）
単語 trigram の perplexity	4.0
単語 bigram の perplexity	13.9
フロアリングの値	exp(-1000.0)
言語尤度と音響尤度の	1
結合値 α	

7.1.3 実験結果

ここで提案したアルゴリズムは HP735、語彙数 1567、ビーム幅 4096 において、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。実験結果は文認識率と単語正解率 (Word correct) と単語認識精度 (Word accuracy)[10] で評価した。また比較のために単語の bigram を使用したときの実験も行なった。実験結果を表 7.2 に示す。実験の結果、特定話者認識において trigram を用いたとき、文認識率で 66.7%、8 位までの累積認識率で 75.1% が得られた。しかし、不特定話者認識では、テストデータ全てにおいて、データの先頭のポーズ区間に 1 音節の単語が挿入されたため、文認識率は 0.0% になった。（例えば「はい」を「と、はい」と認識。）したがって、認識精度が正解率と比較して大きく低下している (31.1% ← 74.2%)。

表 7.2: 認識実験の結果　文認識率 (%)

model		bigram		trigram	
		特定話者	不特定話者	特定話者	不特定話者
累積文認識率	1	42.5%	0.0%	66.7%	0.0%
	~2	47.9%	0.0%	72.4%	0.0%
	~8	51.3%	0.0%	75.1%	0.0%
word-correct		80.7%	55.8%	88.8%	74.2%
word-accuracy		63.0%	1.2%	81.1%	31.1%

text-closed; ビーム幅:4,096; α:1

表 7.3に、特定話者で単語 trigram を使用したときの誤認識の例を示す。例文においてアンダーラインは誤認識を示す。誤認識された文の中には、意味的には正しい文が多い。意味的に正しい文を正解に含めた時、1 位文理解率は約 80% であった。

表 7.3: 実験において誤りが出力された文の例
text-closed; ビーム幅:4,096; α:1

正解文 → 1 位出力
京都プリンスホテルが会議場には近いのですが → 京都プリンスホテルが会議場には近いんですが ホテルの手配もしていただけるのですか → ホテルの手配もしていただけるんですか
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 ご住所とお名前をお願いします → ご住所とお名前_お願いします
住所は東京都港区新橋１丁目１番３号です → 住所は東京都になったのを送っしかし去年一番可能です 電話番号は３３１の２５２１です → 論文を発表３３１の２２日です

7.2　ポーズの処理

表 7.3において、入力された文と大きく異なる文が出力された音声データを調べると、ポーズの区間から誤りが始まっていることがわかった。そこで言語モデルにおいてポーズのスキップ、音響モデルにおいてポーズの HMM の学習をすることで認識性能の向上を試みた。

7.2.1　ポーズのスキップ (言語モデルにおける処理)

ポーズは、文節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 [25]。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。ここで使用したアルゴリズムでは、各時刻・各状態において累積尤度が最大の単語

列を知ることができる。そこでポーズを1単語と考えて、ポーズに接続されたときの連鎖確率値は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の連鎖確率値を計算する。例えば「“東京都” “港区” “新橋” /pause/ “1丁目”」と発声されたとき、単語trigramの値を $P(\text{“新橋”}\,|\text{“東京都”},\text{“港区”}) \times 1.0 \times P(\text{“1丁目”}|\text{“港区”},\text{“新橋”})$ と計算する。

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7.1と同一である。この結果を表7.4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(66.7% → 71.6%)。また、不特定話者認識では、認識性能が顕著に向上した(0.0% → 61.7%)。

表7.4: 認識実験の結果（ポーズのスキップ）　文認識率(%)

model		bigram		trigram	
		特定話者	不特定話者	特定話者	不特定話者
累積文認識率	1	49.4%	31.4%	71.6%	61.7%
	~8	60.2%	44.4%	80.0%	76.7%

text-closed; ビーム幅:4,096; α:1

7.2.2 ポーズのHMMの学習（音響モデルにおける処理）

不特定話者認識の実験において誤認識された文を調べると、テストデータの先頭のポーズ区間から誤認識している例が多いことがわかった。そこでテストデータの先頭の無音区間を利用して、Baum-Welchアルゴリズムでポーズの HMMを再学習した。学習にはテストデータ100文の先頭の100msを使用した。

7.2.3 ポーズ処理をしたときの実験の結果

上記に示すような改良をして、文認識実験を行なった。この実験結果を表 7.5 に載せる。これからわかるように認識性能が向上する。特に不特定話者認識においては効果が著しい。特定話者認識における誤認識の例を表7.6に載せる。これからわかるように、誤認識された文には意味的に合っている文が多い。意味的に正しい文を正解に含めたとき1位理解率は99%に達した。

これらの実験から、誤認識の原因になってい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を学習することが有効であることが示された。

表7.5: 認識実験の結果（ポーズのスキップ、ポーズ学習）　文認識率(%)

model		bigram		trigram	
		特定話者	不特定話者	特定話者	不特定話者
累積文認識率	1	60.5%	44.8%	90.4%	83.9%
	~2	68.2%	51.0%	95.4%	92.7%
	~8	76.2%	55.6%	97.7%	96.6%
word-correct		87.2%	72.4%	97.6%	96.2%
word-accuracy		79.6%	58.3%	97.1%	95.7%

text-closed; ビーム幅:4,096; α:1

表 7.6: 実験において誤りが出力された文 （ポーズのスキップ、ポーズ学習）
text-closed; ビーム幅:4,096; α:1

正解文 → 1位出力
京都プリンスホテルが会議場には近いのですが → 京都プリンスホテルが会議場には近いんですが ご住所とお名前をお願いします → ご住所とお名前_お願いします
ではお名前とご住所をお願いします → ではお名前とお住所をお願いします 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
失礼します → そうします 言語学や心理学を専攻する方にも参加していただく予定です → 言語学や心理学を専攻する方にご参加してあるというんです

7.3 各種パラメータの検討

7.3.1 ビーム幅

ビームサーチは、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ことを仮定している。そこで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 7.1 と同一である。また、 7.2.1 節および 7.2.2 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 7.1 に示す。

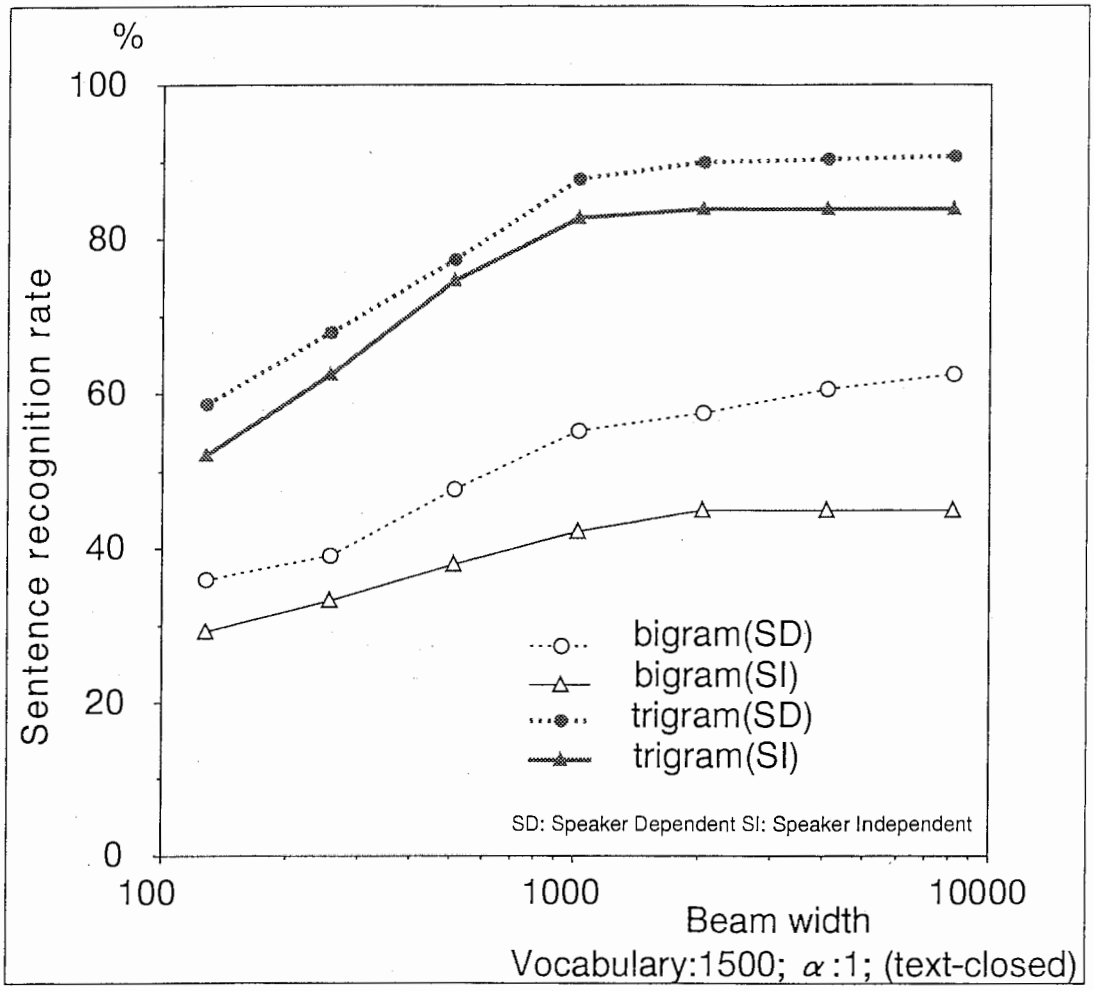


図 7.1: ビーム幅を変化させたときの変化　文認識率 (%)
Sentence recognition rate versus beam width

この実験結果からビーム幅を広げるに従い認識性能は向上するが、ビーム幅が 1024 を越えると、認識性能はあまり変化しないことがわかる。ここでは認識語彙数を変化させた実験を行っていないため明確にはいえないが、このビーム幅 1024 は語彙数 1567 に近いことから、朗読発話においてビーム幅は語彙数程度、必要であると考えている。

7.3.2　音響尤度と言語の連鎖確率の結合値 α

ここでは音響尤度と言語の連鎖確率の結合値 α を変化させたときの文認識率の変化を調べた。他の実験条件は表 7.1 と同一である。この結果を図 7.2 に示す。この図において横軸は結合値 α で、この値が大きいほど言語尤度の重みが音響尤度と比較して増加することを意味している。縦軸は文認識率である。

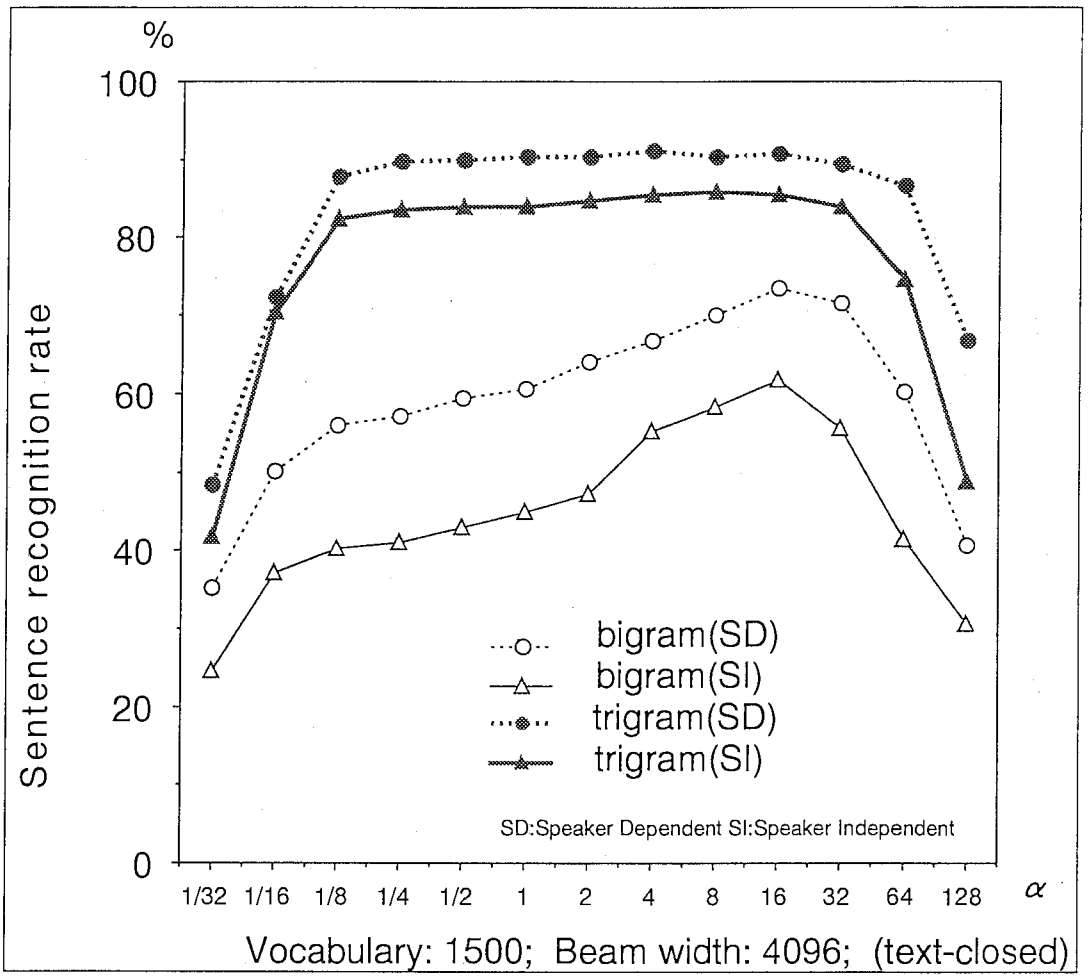


図 7.2: 音響尤度と言語の連鎖確率の結合値を変えたときの認識性能の変化　　文認識率 (%)
Sentence recognition rate versus language weight

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が得られた。ただし、個人的には音響尤度と言語の連鎖確率の結合値 α は 1 が妥当であると考えている。

7.3.3 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 7.1 と同一である。また 7.2.1章および 7.2.2章で述べたポーズ処理はおこなった。

この実験結果を図 7.3 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。

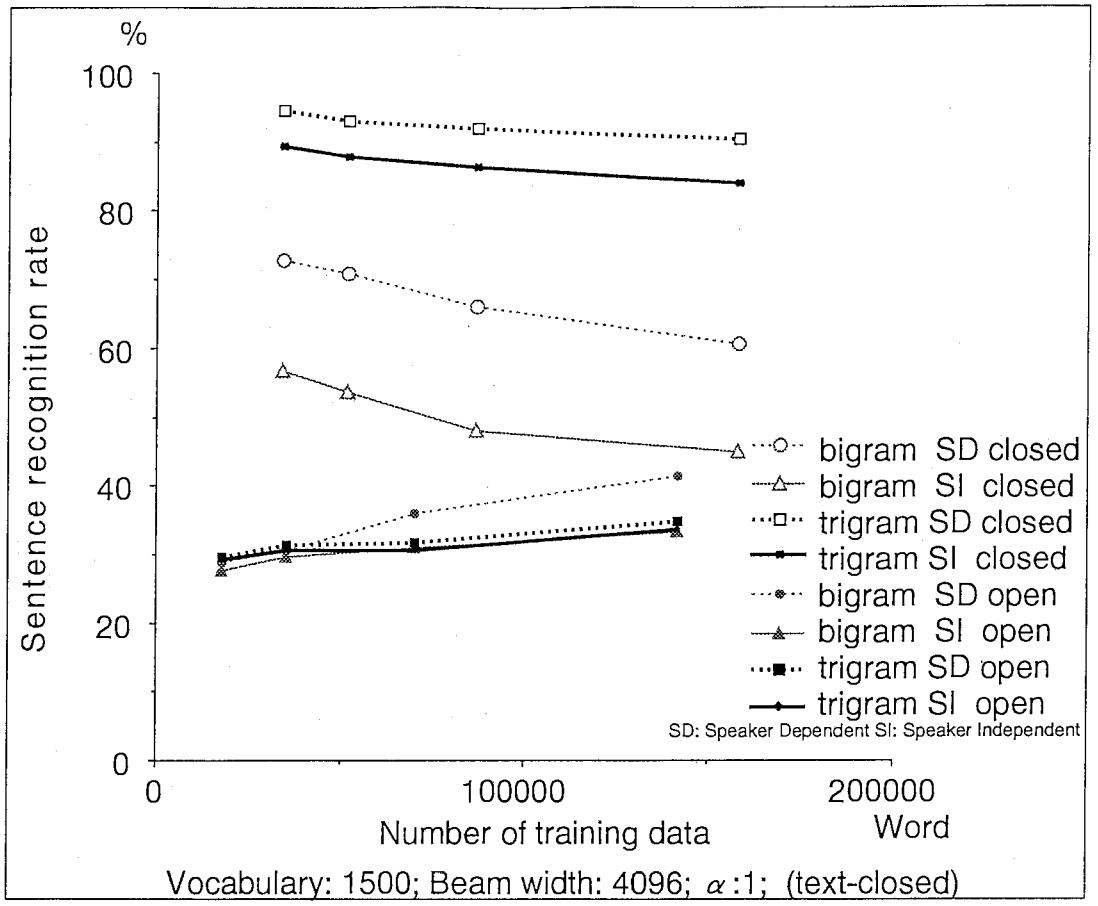


図 7.3: 学習データ量における認識結果の変化　認識率 (%)

7.3.4　単語の trigram の値を平滑化した場合の認識率

単語の trigram の値に deleted-interpolation を利用して平滑化した場合の認識率の変化を図 7.7に示す。なお、平滑化の値は、 trigram,bigram,unigram, フロアリングに対して各々 $\lambda_3 = 0.35, \lambda_2 = 0.48, \lambda_1 = 0.11, \lambda_0 = 0.06$ である。これから単語の trigram を平滑化することで text-open data において認識性能が向上することがわかる。

表 7.7: 認識実験の結果（単語の trigram の値を平滑化したとき）　文認識率 (%)

		text-open		text-closed	
		特定話者	不特定話者	特定話者	不特定話者
base-line	1	35.6%	33.7%	90.8%	85.1%
	~2	37.5%	36.8%	96.6%	93.5%
	~8	38.3%	37.9%	98.8%	97.7%
interpolation	1	51.7%	43.3%	79.3%	78.2%
	~2	58.6%	47.9%	88.5%	86.2%
	~8	62.4%	53.6%	91.9%	90.0%

text-closed; ビーム幅:4,096; α:1

7.4 考察

7.4.1 フルサーチと Viterbi サーチ

Viterbi サーチ (one-pass DP) は連続音声認識アルゴリズムとして知られている。このアルゴリズムにおいて各単語の HMM の最後の状態と後続する単語の最初の状態の遷移において単語の trigram の連鎖確率値を掛けることによって最尤の単語列を選出できる [38]。ただし最尤解を保証するには、現在の単語と 1 つ前の単語を記録しながら最大累積尤度を記録する必要がある。したがって、この記憶容量 (通常 grid と呼ばれる) は O(語彙数^2) 必要である。一方フルサーチの記憶容量は O(語彙数$\hat{}$文が構成する単語数) 必要である。したがって両者ともビームサーチが必要になる。ここでは N-best リストが利用できるフルサーチを利用した。

7.4.2 ポーズの HMM の学習に関して

本実験では、ポーズの HMM は Baum-Welch アルゴリズムを用いて再学習をおこなった。しかし、データ量が少ない場合のことを考えると、混合分布の平均値を移動させる話者適応のアルゴリズム [47] を使用したほうが好ましいと考えている。

7.4.3 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。今後、ポーズは促音やクロージャとも併せて考慮する必要があろう。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。

7.4.4 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。

7.4.5 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 (図 7.4)。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 であると考えている。

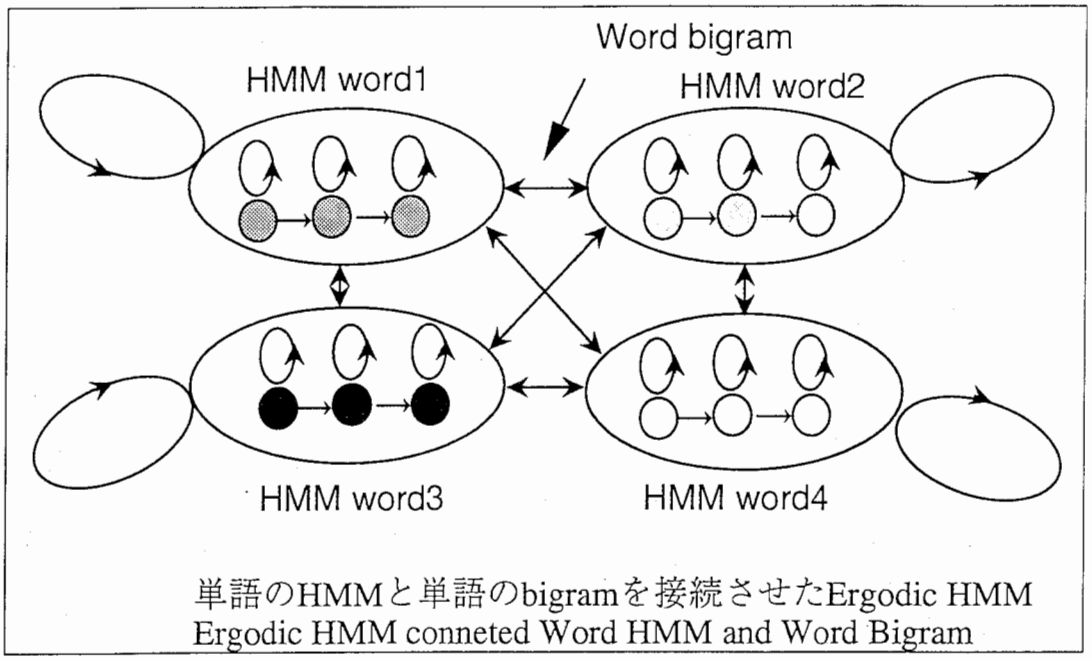


図 7.4: bigram と HMM を組み合わせた Ergodic HMM
Ergodic HMM with word HMM and word bigram

7.5 まとめ

本論文では、単語 trigram を利用した実験結果を報告した。実験の結果、朗読発話の text-closed data において特定話者認識では 66.7% の文認識率が得られた。 この論文ではフルサーチを利用している。したがって、各時刻・各状態において累積尤度が最大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削減できる。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。このようなポーズの処理をすることにより不特定話者認識の text-closed data において 83.9% の文認識率が得られた。 これらの実験の結果、このアルゴリズムの有効性が示された。

第 8 章

自由発話の音声認識アルゴリズム

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」などに代表される間投詞や、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。しかし、このような発話様式では、認識精度の高い音響モデルの作成は困難であると考えられる。そこで認識性能を向上させるため、perplexity の低い言語モデルが必要になる。

現在、音声認識に用いられている言語モデルは、簡潔さ・有効などの点から単語の bigram モデルが主流である [27]。しかし、単語の trigram モデルの perplexity は bigram より一般的に低いことが知られている。そこで、ここでは 4.2章で述べたアルゴリズムを基本に言語モデルとして単語の trigram を用いて自由発話の認識を試みた。

8.1 間投詞や言い直しの対策

また、自由発話においては、「あのー」「えーと」などに代表される間投詞や、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が [35]、この対策方法の 1 つに garbage モデルを使用する方法がある。garbage モデルは、キーワードスポッテングにおいて使用されていたモデルで、キーワード以外の音素を数個の HMM でモデル化しようとするものである [12][63]。これを Viterbi サーチに組み込み、間投詞や言い直しなどの不要語を garbage モデルで対応する方法を井上らは提案している [17]。

この方法は、間投詞や言い直しを音響モデルで解決した方法と言える。しかし、言語モデルにおいて、間投詞や言い直しを音素の系列とみなし、この音素をスキップすることで同様なことが実現できる。本論文では言語モデルに単語の trigram を用いて、この 2 つの方法で自由発話の認識を試みた。

8.1.1 garbage モデル (音響モデルによる対策)

garbage モデルは、間投詞や言い直しを 1 個ないし数個の garbage HMM で処理しようとする方法である。garbage HMM は、あらかじめ全ての音素を学習しておく。そして認識アルゴリズムにおいて 7.2.1 節のポーズの HMM と同様に扱う [16]。この結果、音声データ中の間投詞や言い直しは garbage モデルで認識しながら、言語モデルではこれらの言語現象をスキップすることで自由発話の音声が認識できる [17]。

8.1.2 音素スキップ（言語モデルによる対策）

間投詞や言い直しは、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。そこで、間投詞や言い直しを音素系列として認識しながら、言語モデルでは音素系列をスキップすることにより自由発話の音声が認識できる。ただし、このようなアルゴリズムでは、音声データ全てが音素系列と認識される可能性があるため、本論文ではペナルティとして音素の trigram を使用する。

例えば「“東京都” “港区” “新橋” “あのう (anou)” “1 丁目”」と発声されたとする。そして「あのう」は間投詞とする。

このときの 言語モデルの連鎖確率値は $P($ “新橋” | “東京都”, “港区”$) \times P(/a/|/sh/,/i/) \times P(/n/|/i/,/a/) \times P(/o/|/a/,/n/) \times P(/u/|/n/,/o/) \times P($“1 丁目” | “港区” “新橋” $)$ と計算する。

ここで、$P(/a/|/sh/,/i/)$ はペナルティ、$P($“1 丁目” | “港区” “新橋” $)$ は” あのう” を音素系列と見てスキップしたことを意味する。

この方法は、garbage モデルを言語モデルで実現する方法であるとも言える。また、既に提案されている未知語検出のアルゴリズムと基本的には同一の思想である [5],[23],[19],[22],[30],[11]。ただし、これらの論文では未知語検出を目的にしている。また、使用している言語モデルも異なる。

8.2 自由発話の文認識実験条件

認識実験は、音響モデルには不特定話者の HMM、言語モデルには単語の trigram を使用して行なった。実験条件は表 7.1 とほぼ同じであるが、語彙数やビーム幅などは異る。garbage モデルは、4 状態 3 ループの 10 混合のモデルで、男性話者 12 名の音素バランス 216 単語から作成した。音素の trigram の連鎖確率値は「あのー」、「えーと」などの間投詞を含めて国際会議の予約に関するデータ約 1 万 2 千文章、約 17 万単語から作成した。実験条件を表 8.1 に示す。また全ての実験において 7.2.1 節および 7.2.2 節で述べたポーズの処理を行なっている。

表 8.1: 文音声認識の実験条件

HMM の学習音声	男性話者 12 名の 736 単語発声
garbage モデルの学習音声	男性話者 12 名の音韻バランス 216 単語
garbage モデル	4-state 3-loop 10mixture left-right model
認識単語数	435
ビーム幅	16,384
単語 trigram の値の	約 1 万 2 千文章 171,978 単語
推定に使用した	テストデータのテキストを含む
テキストデータ量	（間投詞は削除）
音素 trigram の値の	約 1 万 2 千文章 171,978 単語
推定に使用した	テストデータのテキストを含む
テキストデータ量	（間投詞を含む）
言語尤度と音響尤度の	16
結合値 α	

8.2.1 自由発話の音声データ

音声データは以下に示すような方法で収録した。ただし、話者は一般人である。

1. 朗読発話

   テキストを読みあげた音声データ。テキストの内容は 7.1.2 節において使用されたテストデータと同一。間投詞や言い淀み・言い直しは無い。このデータは text-closed の実験になる。

2. 疑似自由発話

   間投詞を含むテキストを読みあげた音声データ。間投詞を除いて、「1 朗読発話」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 朗読発話」と異なる。間投詞や言い直しや未知語を含む。このデータは text-open の実験になる。

8.2.2 単語の trigram の平滑化

単語 trigram は語彙数の 3 乗のパラメータの数をもつ。したがって全ての trigram の値を直接推定できるだけの大量のテキストデータを収集することは困難である。そのため、text-open の音声データを認識させる場合、通常 trigram の連鎖確率値は平滑化して使用される。ここでは deleted-interpolation[21] を使用した。そして、単語の trigram の値を平滑化した場合としなかった場合の両方で実験を行った。

8.3 自由発話の文認識実験結果

表 8.2 に単語の trigram の連鎖確率値を平滑化しないで認識実験を行なった結果を示す。また、表 8.3 に単語の trigram の連鎖確率値を deleted-interpolation で平滑化して認識実験を行なった結果を示す。なお、平滑化の値は、trigram,bigram,unigram, フロアリングに対して各々 $\lambda_3 = 0.35, \lambda_2 = 0.48, \lambda_1 = 0.11, \lambda_0 = 0.06$ となった。

表 8.2: 自由発話の文認識実験結果 （平滑化無し）

文認識率 (%)

	累積文認識率	base-line	garbage	skip-phone
朗読発話	1	89.7%	83.2%	88.5%
	~2	97.3%	90.5%	96.2%
	~8	100.0%	97.3%	99.2%
	Word-Correct	97.5%	93.4%	96.4%
	Word-Accuracy	96.9%	93.2%	96.0%
疑似自由発話	1	41.6%	64.5%	73.3%
	~2	43.1%	70.2%	79.0%
	~8	44.3%	78.2%	82.8%
	Word-Correct	70.6%	81.5%	89.5%
	Word-Accuracy	34.2%	76.6%	82.3%
自由発話	1	10.7%	37.8%	47.7%
	~2	15.3%	46.9%	57.2%
	~8	19.5%	56.1%	66.8%
	Word-Correct	44.7%	65.7%	80.9%
	Word-Accuracy	9.1%	58.9%	73.3%

不特定話者認識; 語彙数:435; ビーム幅:16,384; α:16

trigram の連鎖確率を直接使用

表 8.3: 自由発話の文認識実験結果 （平滑化有り）

文認識率 (%)

	累積文認識率	base-line	garbage	skip-phone
朗読発話	1	47.3%	49.2%	46.9%
	~2	52.2%	53.8%	53.4%
	~8	61.5%	61.8%	64.1%
	Word-Correct	77.4%	70.1%	77.1%
	Word-Accuracy	71.9%	68.0%	72.2%
疑似自由発話	1	29.0%	36.3%	28.6%
	~2	30.1%	37.8%	30.9%
	~8	33.2%	42.0%	36.3%
	Word-Correct	63.1%	59.6%	63.2%
	Word-Accuracy	28.8%	44.3%	29.9%
自由発話	1	10.3%	16.4%	10.7%
	~2	14.1%	18.3%	13.0%
	~8	17.5%	22.1%	16.8%
	Word-Correct	51.0%	41.5%	46.7%
	Word-Accuracy	27.9%	26.5%	19.2%

trigram; 不特定話者認識; 語彙数:435; ビーム幅:16,384; α:16
trigram の連鎖確率を deleted-interpolation して使用

これらの実験から以下のことがわかる。

1. 自由発話において、音素スキップの方法を使用した場合も、garbage モデルを使用した場合も、認識性能は向上する。

2. garbage モデルを使用したときと音素スキップの方法を利用したときの認識率を比較すると、音素スキップの方法を利用したときの方が高い認識性能を得ている。

3. trigram の値を平滑化した場合と平滑化しない場合の認識率を比較すると、平滑化をしないほうが高い認識性能を得ている。

4. trigram の連鎖確率値を平滑化をしないで、音素スキップの方法を利用することで、自由発話では 47.7% の文認識率が得られた。また朗読発話でも、この処理を加えることの認識性能の低下は少なかった (89.7% → 88.5%)。

自由発話において音素スキップをしたときの誤認識の例を、表 8.4に載せる。表中の括弧内は、実際の発話内容である。

表 8.4: 実験において誤りが出力された文（自由発話認識）
（ビーム幅:16,384; α:16）

正解文（発声内容）→ 1位出力
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けれど）
→ 会議の宿泊施設についてお尋ねしたいんですよ
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
（えーと、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）
→ 登録を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
ではお名前と住所お願いします
（ではお名前と住所お願いします）
→ ではお名前とご住所をお願いします
会議の参加料について教えていただきたいのですが
（えー、会議の参加料について教えていただきたいのですけれども）
→ 会議の参加料について教えていただけますか
失礼します
（う、失礼します）
→ そうします
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
（えーっと、京都プリンスホテルに8月4日から8日まで
えーっと一人部屋をお取りしました）
→ 国際会議が8月に行われているんでしょうか

自由発話では発話内容が朗読発話と異なっている。しかし、文認識率の計算においては、朗読発話の単語と一致した場合に正解とした。そのため、発話内容と認識結果が合っていても、誤認識とした。したがって実際の認識性能は 47.7% より高い。意味的に正しい文を正解とすると、1位文理解率で約 75%、8位までの累積文理解率は 90% になった。したがって、音素スキップの方法は、自由発話の認識において有効であると考えられる。

8.4 考察

8.4.1 自由発話認識における trigram の平滑化に関して

今回の自由発話の実験では、単語の trigram の値を平滑化をしない方が、deleted-interpolation で平滑化をした場合より高い認識性能が得られた。この原因を次のように考えている。

音声認識において利用される言語モデルは、通常エントロピー（もしくは perplxity）が低くかつカバー率が広いことが要求される。一般に単語の trigram はエントロピーは低いがカバー率も低い。そこでカバー率を上げるために、deleted-interpolation などの平滑化の方法が利用されている。しかし言語モデルのエントロピーは増加する。一方 garbage モデルや音素スキップは、言語モデルで対応出来ない音声を音素で対応するアルゴリズムである。したがって、この方法を利用したばあい間接的に言語モデルのエントロピーは増加する。したがってこれらのアルゴリズムと deleted-interpolation を組合せると、テストデータにおける perplexity は増加する可能性が

ある。そのため、認識性能が低下する。

自由発話では、文字化した文章と発話した音素列の差は朗読発話より大きくなる。例えば「会議にー（い）」と発声している音声を「会議に」と文字化している。また、「あのー」「えーと」などの間投詞や言い直しは対話文の 50% に出現する [35]。 したがって、自由発話の認識では、全ての音素を完全に認識する必要はなくて、意味的に合っている文章を出力すれば十分であると思われる。そして、自由発話の認識において使用される言語モデルには低い perplexity が求められ、言語モデルがカバーできない範囲は garbage モデルや音素スキップで対処するのが妥当であると考えている。

8.4.2 音素スキップと garbage モデルの比較

今回の実験では、音素スキップの方法が garbage モデルより高い認識性能が得られた。これは、言語モデルが適応できない音声区間は garbage モデルよりも音素モデルで認識したほうが認識性能は高くなることを意味している。しかし、この方法は garbage モデルより一般的に広いビーム幅が必要になると考えている。したがって、語彙数が多い場合やビーム幅が小さい場合、garbage モデルのほうが認識性能は高くなる可能性があると思われる。

8.4.3 間投詞の音素に関して

間投詞には従来の音素では表現できない音素がある [35]。例えば「んー」（考え込むとき発声している音）は /N/ あるいは /uN/ の 両者に解釈できる。したがって間投詞に関しては認識単位を単語にするべきであろう。

8.4.4 自由発話の認識に関して

現在自由発話の認識アルゴリズムとしては、garbage モデルなどを使用する方法の他に、キーワードスポッティングを利用する方法や、始めに音素ラティスを作成し次にキーワードを選択する手法 [59][60] などが試みられている。今後自由発話の認識において、これらの方法も考慮する必要があろう。

8.5 まとめ

本論文では、自由発話の認識を行なった。このような発話に特有な間投詞や言い誤りは、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、間投詞や言い誤りを音素の系列と捉え、この音素をスキップをすることにより、文認識率が 10.7% から 47.7% に向上した。そして意味的に正しい文まで正解とすると、1 位文理解率で約 75%、8 位までの累積文理解率は 90% になった。この実験は text-open data の認識実験である。これらの実験の結果、このアルゴリズムの有効性が示された。

第 9 章

結論

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」などに代表される間投詞や、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。本論文では、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を試みた。このような発話様式では、認識精度の高い音響モデルの作成は困難であると考えた。そこで認識性能を向上させるため、perplexity の低い言語モデルと、そのサーチ問題について研究した。

本論文では以下の順序にしたがって自由発話のための音声認識アルゴリズムについて述べた。

1. 言語のマルコフモデルにおけるエントロピーと収束性

   言語をマルコフモデルで表現するときのデータ量と収束性について調べた。その結果、98% は、ほぼマルコフモデルで近似できるが、のこり 2% が収束しないことを述べた。

2. 自由発話の音響的、言語的特徴

   ここでは自由発話と朗読発話の違いについて調べた。この結果、アナウンサーでは、両者に大きな違いは無いことが示された。

3. 連続音声認識のアルゴリズム

   ここではフルサーチと One-Pass サーチの違いについて述べた。そして One-pass サーチはフルサーチにおいてローカルビームをとっているにしか過ぎないこと、そしてグリッドを中心に考えれば、基本的には両者を統一できることを述べた。

4. シュミレーションによる音節や漢字の trigram の有効性

   ここでは入力段において音節マトリックスを想定した場合の、trigram の有効性について述べた。この結果、データが大量にあれば text-open data でも、高い認識性能が得られることが示された。

5. 単語の HMM と単語の bigram を用いた文節認識アルゴリズム

   医療 WP のための音声認識システムの詳細なアルゴリズムとその実験結果である。この実験から単語 HMM と単語 bigram でもかなり高い認識性能が得られること、単語 HMM は Fuzzy-VQ HMM を使用することにより、1 回の発声でも、学習が可能であることを示した。

6. 単語の trigram を用いた文認識アルゴリズム

   ATR の国際会議の予約のタスクにおいて、連続分布 HMM と単語 trigram を使用した認識アルゴリズムとその実験結果について報告した。このアルゴリズムではフルサーチを使用

している。そして認識性能を向上させるためにポーズのアダプテーションおよびスキップが重要であることを示した。

7. 自由発話認識のためのアルゴリズムとその実験結果

最後に自由発話を試みた。対話文の 50% は「あのー」、「えーと」などの間投詞を含む。また、言い直しは約 10% に出現する [35]。これらの間投詞や言い直し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。そこでこれらの単語をポーズ処理と同様にスキップすることで、自由発話の認識が可能になる。最後に、このアルゴリズムと認識実験結果について述べた。そして、これらのアルゴリズムの有効性について述べた。

第 10 章

謝辞

この研究にあたって多くの人の協力を得ました。

新聞記事の解析には日本文訂正支援システムの辞書を使用した。これらの辞書は宮崎正弘新潟大教授、安田研究員、高木、島崎研究員等の方々と池原が開発したものを使用させていただきました。また、認識実験において用いたHMMの特定話者モデルは現在シャープ株式会社の山口耕一氏から、不特定話者モデルは小坂哲夫氏から頂きました。また、磯谷亮輔氏にはdeleted-interpolationの値や単語のtrigramを用いたViterbiサーチのアルゴリズムに関してコメントを頂きました。自由発話の言語のデータベースは、匂坂氏や現ＮＨＫの江原氏の指示のもとに作成されました。また、ATR音声翻訳通信研究所の森元室長や飯田室長の他、各研究員に多くの協力をいただきました。そして、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には研究の機会を与えて頂きました。さらに音声翻訳通信研究所の第一研究室の方々には熱心な御討論と有益な御助言をいただきました。これらの皆様に感謝致します。

第 11 章

ATR および「研究」に対する感想

11.1 「研究」の感想

ここでは研究そのものの感想を書く。

1. 研究の評価に関して

   研究の評価は人によって大きく異なる。例えば、言語をマルコフモデルで表現することを、とんでもないバカな発想であると考える人が多い。しかし、認識性能が簡単に向上する以上、優れた発想であると思う人もいる。基本的には研究の評価は他人によってなされる。しかし、最後は本人しか評価できないと感じている。

2. 論文の査読について

   会議もしくは論文では査読があるが、査読者にとって理解しやすい（査読者の主義、主張に反しない）論文が通りやすく、そのため仲間内の同窓会になっている傾向があるような気がする。

3. 論文の価値について

   論文は、人に読まれてこそ価値がある。したがって引用されるような論文を書くべきである。個人の主義主張のみを書く論文は、書くべきではない。

4. 論文の書き方について

   論文を書くための大原則は以下の通りである。

   (a) 他人の論文を紹介する。

   (b) この論文において、著者が考えた問題点と解決方法を述べる。

   (c) この解決方法に従って実験をする。

   (d) 解決方法の有効性を示す。

   この大原則に従わない論文が多過ぎる。酷い場合には「自己宣伝」としか感じないばあいもある。

11.2 ATR の感想

ATR に 1991 年 3 月から 1994 年 3 月まで在籍した。この感想を述べる。

11.2.1 ATRの長所

1. 多くの人の本音が聞けた。

   個人的に、この研究に関してどのように思っているか、なぜ、そのように感じているか、などの情報は、かなり貴重である。しかし、これらの情報は学会などでは聞けない。やはり、ATRでは本音を聞けるかなり良い機会であったと思う。

2. 機械の設備やデータベースが、かなり揃っていた。

   ATRにおける機械やデータベースは、日本国内、いや世界的に見ても、かなり高水準である。音声処理や言語処理では、あまりお金にならないため、各企業ではATRなみに設備を整えるのは不可能であろう。

11.2.2 ATRの問題点

ATRの問題点は多い。気になる点のみを挙げる。

1. 研究目的

   研究目的が明確でない。自動翻訳のための研究ならば、もっと異なった手段があると思われる。

2. データ依存性

   全てはデータに依存する。手法（アルゴリズム）はデータから推定される。データが決まらなければ、手法は決まらない。この大原則が無視されている。

3. 論文の引用

   論文の大原則に従わない論文が多過ぎる。特に、論文のオリジナリティを出すために「1）他人の論文を紹介する。」において、都合のよい論文しか載せていない場合が多い。1つの論文において多くの見方があるため、ある程度は仕方ないが、その基準を越えている場合が多い。酷い場合には「自己宣伝」としか感じないばあいもある。そのためか、ATRが出した論文も、引用されない論文が多いと感じている。

4. 謝辞に関して

   他人が作成したデータやプログラムは最低限acknowledgeに書くべきである。これが守られていない。また自分の論文のオリジナリティの主張も曖昧（自己満足）である。

5. コンピュータに関して

   無駄に使用されていたコンピュータも多い。また使用されないデータベースも多い。

11.3 各研究に対する筆者の個人的な評価

1. 言語モデルに関して

   自然言語のアルゴリズムは、基本的に計算機言語学の発達に依存してきた。しかし、自然言語は計算機言語とことなり、常に曖昧性を含んでいる。したがって、言語をルールで記述することが可能であろうか？私には、不可能としか思えない。また、人工知能の歴史を振り返えると、CHESSマシンが強くなったのは、ルールを入れることではなく、確率をいれることであった。一見、ルールにしたがって動いているようにみえるCHESSでも、確率を入れなければ強いプログラムにならないということは何を意味しているのだろうか？

2. それでも言語はマルコフだ。

言語をマルコフモデルで表現することを、とんでもないバカな発想であると考える人が多い。しかし、実際に現在のコンピュータの能力では、マルコフモデルがもっとも強力なモデルである。たしかに言語現象を考慮したとき、マルコフモデルでは表現できないことも多い。しかし、たとえば文脈自由文法を考えると、カバー率は高いかも知れないが、同時に多くの非文が生成されてしまう。したがってデータをみて、それから最適なモデルを選択するべきである。また、「言語モデルをマルコフモデルで表現した場合、text-closed data に対しては認識性能は良くなるが、text-open data に対しては全然認識できない。」といっている人がいるが、これは大きな誤りである。ATR のデータベースの収集の仕方があまりにも拡散して収集しているため、このような現象がおきるのであって、分野が限定されているばあい、（例えば X 線 CT 所見や新聞記事）約 100 万単語程度収集すれば問題がない。そして収集仕切れないデータに関しては音素スキップ等の一種の grabage モデルで対応できると思っている。

なお、先駆者は常に無視される。

3. 文法に関して

通常、言語には文法があると考えられている。しかし、私の考えでは、文があって文法があると考えている。また、文は無限大に存在する。したがって明確な文法は存在しないと考えている。文法は文に対して十分条件か必要条件かを十分に考える必要がある。また、コンピュータはプログラムされたこと以外はできないことも十分考慮すべきである。

4. 音響モデルに関して

音響モデルは、基本的に音素の単位で認識をしている。しかし、認識をしたいのは音素ではなく、単語である。したがって認識単位は単語で評価するべきだと思う。

5. 韻律情報に関して

多くの言語研究者は韻律情報を音声認識に使用したいと考えている。しかし個人的には、韻律のもつ情報は、かなり少ないと考えている。具体的には日本語の 1 音節ぐらいの情報しか持たないと考えている。この程度の情報量では認識性能はあまり向上できないと思っている。

6. ATR のデモシステムに関して

ATR のデモシステムは、音素同期型の SSS+LR で動作している。しかし、特定話者、文節認識、である。一方フレーム同期の Normal HMM+word bigram を使用することにより不特定話者、文認識が可能である。今後デモシステムは、こちらにするべきであろう。

7. Simple is best

全ての物事は"Simple is Best" である。問題は、何に対して Simple であるかである。これはデータに依存する。

したがって論文では、何を解こうとしているのか、それに対し、最も simple な方法はなにか？自己が使用としているのは、これに比べてどのように違うのかを主張する必要がある。これがされているように感じない。

8. 研究のオリジナリティ

私のオリジナリティを述べる。ただし、全ての論文を見ることは不可能であるため他に先行研究がある可能性も高い。

(a) 音節および漢字かなの trigram の有効性の主張 (1988). この論文で音声認識において音節および漢字かなの trigram が有効であることを述べた。

(b) ポーズおよびアクセント位置の情報量 (1988) ポーズおよびアクセント位置の情報が日本語の 1 音節程度であることを述べた。

(c) 漢字かな文字の trigram による仮名漢字変換 (1991). 漢字仮名文字の trigram を使用しても仮名漢字変換率が高いことを述べた。

(d) 漢字かな文字の trigram による形態素解析 (1992). 漢字仮名文字の bigram,trigram を使用して十分な形態素解析が得られることを述べた。

(e) 自由発話と朗読発話の音響的モデルの違い (1991). 少なくともアナウンサーの発声では音響モデルでは自由発話と文の朗読発話に差がないことを述べた。

(f) フルサーチによる連続音声認識アルゴリズム (1994). フルサーチは、大量のメモリが必要であるが、ビームサーチを使用することによりあまり問題がないことを述べた。

(g) One-pass における N-best の認識アルゴリズム (1991). One-pass サーチにおいてグリッドを複数候補もたせることにより N-best が出力できる。この認識アルゴリズムを書いた。

(h) フレーム同期型の音声認識 (1991). これは世界的には当たり前であるが ATR では、私が始めてアルゴリズム書いた。

参考文献

[1] 荒木 哲朗, 村上 仁一, 池原 悟,“m 重マルコフモデルを用いた音 節ラティスからの候補絞り込みアルゴリズム”, 電子情報通信学会技術報告, CS90-55, (Dec. 1990).

[2] 荒木 哲朗, 村上 仁一, 池原 悟,“二重音韻マルコフモデルによる日本語の文節音韻認識候補の曖昧さの解消効果”, 情報処理学会論文誌, Vol.30, No.4, pp.467-477 (1989.4).

[3] X.D.HUNAG, Y.Ariki, M.A.Jack, “Hidden Markov Models for Speech Recognition”,EDINBURGH UNIVERSITY PRESS (1990).

[4] 有田 英一, 小暮 潔, 野垣内 出, 飯田 仁, “メディアに依存する会話の様 式”, NL61-5, (1987).

[5] A.Asadi,R.Schwartz and J.Makhoul, “ Automatic Modeling for Adding New Words to a Large Vocabulary Continuous Speech Recognition System”, ICASSP91, (1991).

[6] A.Averbuch, “An IBM PC BASED LARGE-VOCABULARY ISOLATED-UTTERANCE SPEECH RECOGNIZER”, Vol.1, 2.4.1, PP.53-56 ICASSP86, (1986).

[7] 江原 暉将, 小倉 健太郎, 森元 逞, “電話対話データベースの構築”, 第 40 回 情報処理全国大会, pp.486-491 (1990).

[8] G.D.Forney, ”the Viterbi Algorithm”, Proc.of IEEE, Vol.61, pp.268-278 (1973).

[9] 古井 貞熙,“日本語単音節音声認識の検討”, 信学会全大, No.1351, pp.5-329(1981).

[10] Cambridge University Engineering Department Speech Group and Entopic Research Laboratories Inc., “HTK:Hidden Markov Model Toolkit V1.5” (23 Sepember 1993).

[11] 花沢利行, 中島邦男, “音声タイプライタを用いた未知語検出方法の改良検討”, 日本音響学会平成 4 年秋季研究発表会講演論文集, pp.219-220, (Oct. 1992).

[12] Higgins A.L. and Wohlford R.E.:”Keyword recognition using template concatenation”, Proc.ICASSP85,pp. 1233-1236 (March 1985).

[13] 飯田 仁, 野垣内 出, 相沢 輝昭, “通訳を介した電話対話の特徴分析”, 電子通信学会技術速報, NLC86-11 (1986).

[14] 池原悟, 安田, 島崎, 高木:”日本文訂正支援システム REVISE） ”, 研究実用化報告,Vol.36,No.9,pp.1159-1167,1987

[15] 池原悟, 白井諭:”単語解析プログラムによる日本文誤字の自動検出と二次マルコフモデルによる訂正候補の抽出 ”, 情処論,Vol.25,No.2,pp.298-305(1984.3)

[16] 今村 明弘、北井幹雄, "事後確率を用いたフレーム同期型ワードスポッティング ," 日本音響学会平成5年度秋季研究発表会, 1-4-2, pp.3-4 (Mar. 1993).

[17] 井ノ上直己, 武田 一哉, 山本誠一, "Garbage HMM を用いた自由発話文中の不要語処理手法", 電子情報通信学会論文誌,A Vol.J77-A, No.2, pp.215-222 (1994.2).

[18] 田中 和世, 板橋 秀一, 他, "音声の知的処理に関する調査報告書", システム技術開発調査研究 3-R-2, 財団法人 機械システム振興協会, (平成4年3月)

[19] 伊藤 克亘, 速水 悟, 田中 穂積, "連続音声認識における未知語の扱い", 電子情報通信学会 技術報告,SP91-96 (1991-12).

[20] 伊藤, 中川 聖一:"確率オートマトンと品詞の3字組出現確率を用いた文節音声認識 ", 音講論集,pp.145 鴫 146 (1988.10)

[21] F.Jelinek, "Self-Organized Language Modeling for Speech Recognition", Readings in Speech Recognition, Morgan Kaufmann Publishers, Inc. San Mateo, California pp.450-506.(1990)

[22] 甲斐 充彦, 中川 聖一, "日本語連続音声認識システム SPOJUS-SYNO の改良と評価", 電子情報通信学会技術報告,SP93-20 pp.49-56 (1993-06).

[23] 北 研二, 江原 暉将, 森元 逞, "連続音声認識における未知語 処理", 日本音響学会講演論文集, 3-5-3, pp.93-94 (Mar. 1991)

[24] 小林 聡, 山本 幹雄, 中川 聖一, "間投詞・言い直し等の出現に関する音響的特徴", 情報処理学会, 音声言語処理研究グループ資料 93-SLP-1-2, pp.7-10 (1993).

[25] 小林 聡, 甲斐、山本 幹雄, 中川 聖一, "間投詞の出現位置の特徴分析と音声認識システムの評価", 情報処理学会, 音声言語処理研究グループ資料, 92-SLP-3-4 (1992).

[26] Kucera.H.,Francis, W.N., "Computational Analysis of Present-day American English," Brown University Express. Providence, Rhode Island. (1967).

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

[28] G.Maltese, F.Mancini, "An Automatic technique to include grammartical and Morphological information in a trigram-based statistical language model", ICASSP92, Vol.1, pp.157-160, (1992).

[29] 松永 昭一, 好田: "branch&bound 法の効果と Bottom-up 音節認識を利用した候補選択", 音声研資 S85 鴫 79, pp.611-620

[30] 南 泰浩, 山田智一, 鹿野清宏, "番号案内を対象とした大語彙連続音声認識アルゴリズム ", 電子情報通信学会技術速報, SP92-108 (Dec. 1992).

[31] 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).

[32] 宮崎正弘,"係り受け解析を用いた複合語の自動分割法 ", 情処論,Vol.25,No.6,pp.970-979(1984.11)

[33] 宮崎正弘、大山芳史“日本音声出力のための言語処理方式,” 情処論文誌,Vol.27, No.11, pp.1053-1061 (1986).

[34] 村上仁一,“メモリ量および計算量を削減した Baum-Welch アルゴリズムの提案と言語モデルへの適用,” 音学講論,1-Q-6,153-154 (1994-10) .

[35] 村上仁一, 荒木哲朗, 池原悟, “日本文音節入力に対して 2 重マルコフ連鎖モデルを用いた漢字かな交じり文節候補の抽出精度,” 信学会論文誌, D-2, Vol.J75-D-2, No.1, pp.11-20 (Jan.1992).

[36] 村上仁一, 嵯峨山茂樹, “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”, 電子情報通信学会技術報告 SP91-100, pp.71-78 (Nov. 1991)

[37] 村上仁一, 坪井俊明, “BIGRAM をもちいた音節 HMM による文節音 声認識,” 音響学会講演論文集, 3-5-15, pp. 117-118 (1991-04).

[38] 村上仁一, 松永昭一,“単語の trigram を利用した文音声認識アルゴリズムの改良と、非朗読発話認識への拡張”, SP93-127, pp.71-78 (1994-01).

[39] 村上 仁一, 荒木 哲朗, 池原 悟, “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).

[40] 村上 仁一, 荒木 哲朗, 池原 悟,”二重マルコフ音節連鎖確率を使用した単音節音声入力の改善”,SP-88-29,pp.63-70 (1988.6)

[41] 永井 明人, 他, “HMM-LR 連続音声認識装置の開発と性能評価”, 日本音響 学会平成 3 年度秋季研究発表会, 1-5-23, pp.45-46, (Oct. 1991).

[42] 長尾 真, “日本語情報処理”, 信学誌,1984

[43] 中川聖一, “確率モデルによる音声認識”, 電子情報通信学会,1988.

[44] 中川聖一, “音声入力を想定したあいまいな発話文の理解システムに関する研究”, 文部省科学研究費補助金, 一般研究 (B) 研究成果報告書, pp.137-144, (平成 6 年 3 月) .

[45] C.Nakatani and J.Hirschberg, “A Speech-First Model for Repair Detection and correction”, In Proceedings 31st Annual meeting of theAssociation for computational linguistics, pp.46-53 (1993).

[46] 岡田 美智男, “文脈自由な句構造文法による One-Pass DP 法の構文制御について”, 日本音響学会平成 2 年春季研究発表会講演論文集, pp.91-92, (Mar. 1990).

[47] 大倉 計美, 杉山 雅英, 嵯峨山 茂樹, “混合連続分布 HMM を用いた移動ベクトル場平滑化話者適応方式” 日本音響学会平成 4 年度春季研究発表会, 2-Q-17, pp.191-192 (Mar. 1992).

[48] 佐川雄二、大西昇、杉江昇、 “対話文における誤りの自動修復”, 情報処理学会 自然 言語処理研究会資料, 93-10, pp.71-78 (Jan. 1993)

[49] 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).

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

[51] 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).

[52] 金田一京助, 他, "新明解 国語辞典 第3版", (1987-10).

[53] 篠崎 直子, 小倉 健太郎, 森元 逞, "言語データベース作成のためのシュミ レーション会話", 第37回情報処理全国大会, pp.1000-1001 (1988).

[54] E.Shriberg, J.Bear and J.Dowding,"Automatic detection and corrction of repairs in Human-Computer Dialog", Proc. Speech and Natual Language Workshop, pp.419-424 (1992).

[55] 高木 一幸, 保浦 直子, 板橋 秀一, "対話における話題展開と発話単位の性 質", 情報処理学会, 音声言語処理研究グループ資料, 93-SLP-1-3 pp.11-18 (1993).

[56] 武田 一哉, 匂坂 芳典, 片桐 滋, 桑原 尚夫,"音韻ラベルの持つ日本語音声 データベースの構築", SP87-19 (1987).

[57] 坪井 俊明、管村 昇、富久、小橋:"文節発声の日本語入力システムにおける日本語変換法 ", 信学会論文誌,D-2, Vol.j72-D-2,No8,pp.1284-1290(1989.4)

[58] 坪井 俊明, 菅村 昇 "文章作成支援装置の評価," 電子情報通信学会技 術報告,SP90-36, PP.17-22 (1990.8).

[59] 坪井 宏之, 橋本 秀樹, 竹林 洋一, "連続音声理解のためのキーワードラティスの解 析," 日本音響学会平成3年度秋季研究発表会, 1-5-11, pp.21-22 (Oct. 1991).

[60] Wayne Ward and Sunil Issar, "Recent Improvements in the CMU Spoken Language understanding system", ARPA HUMAN LANGUAGE TECHNOLOGY WORKSHOP, pp.208-211, (Mar. 1994).

[61] 渡辺 隆夫, 塚田 聡, "音節認識を用いた尤度補性による未知発話のリジェクション", 電子通信学会論文誌, D-2 Vol.J75-D-2 No.12 pp.2202-2009 (1992-12).

[62] 渡辺 隆夫, 畑崎:"音節をベースとする日本語音声認識 ", 音声研資,S85 鴫 62,pp.477-484

[63] Wilpon G., Lee C. and Rabiner R.: " Application of Hidden Markov Models for Recognition of a Limited Set of Words in Unconstrained Speech", Proc. ICASSP89, pp.254-257 (May 1989).

[64] Xuedong Huang, et.al "An Overview of the SPHINX-2 Speech Recognition System", ARPA Human Language Technology Workshop (Mar. 1993).

[65] 山本 幹雄, 小林 聡, 中川 聖一, "音声対話文における助詞落ち・倒置の分 析と解析手法", 情報処理学会論文誌 Vol.11, No11, pp.1322-1330 (1992).

[66] 吉本啓, " 日本語品詞の分類", ATR Techinical Report TR-I-0008 (Nov,1987).

[67] Victor Zue, Et al " Pegasus: A Spoken Language Interface for On-Line Air Travel Planning", ARPA HUMAN LANGUAGE TECHNOLOGY WORKSHOP, pp.196-201, (Mar. 1994).

[68] Victor Zue, James Glass, David Goodine, et al. "The MIT ATIS System: Preliminary Development, Spontaneous Speech Data Collection,And Performance Evaluation", Eurospeech 91, pp.537-540, Genova, Italy, (Sep. 1991)

[69] Victor Zue, Nancy Daly, et al, "The Collection and Preliminay Analysis of a Spontaneous Speech Database", DARPA Workshop 1989, pp.126-134 (1989).

[70] 徳永 豪, "ランダムアルゴリズムの話題から", 電子情報通信学会誌, Vol.77, No9, pp.957-967 (1994-9).

[71] 青江 順一, "静的ハッシュ法とその応用", 情報処理, Vol.33, No 11, pp.1359-1366 (1992-11).