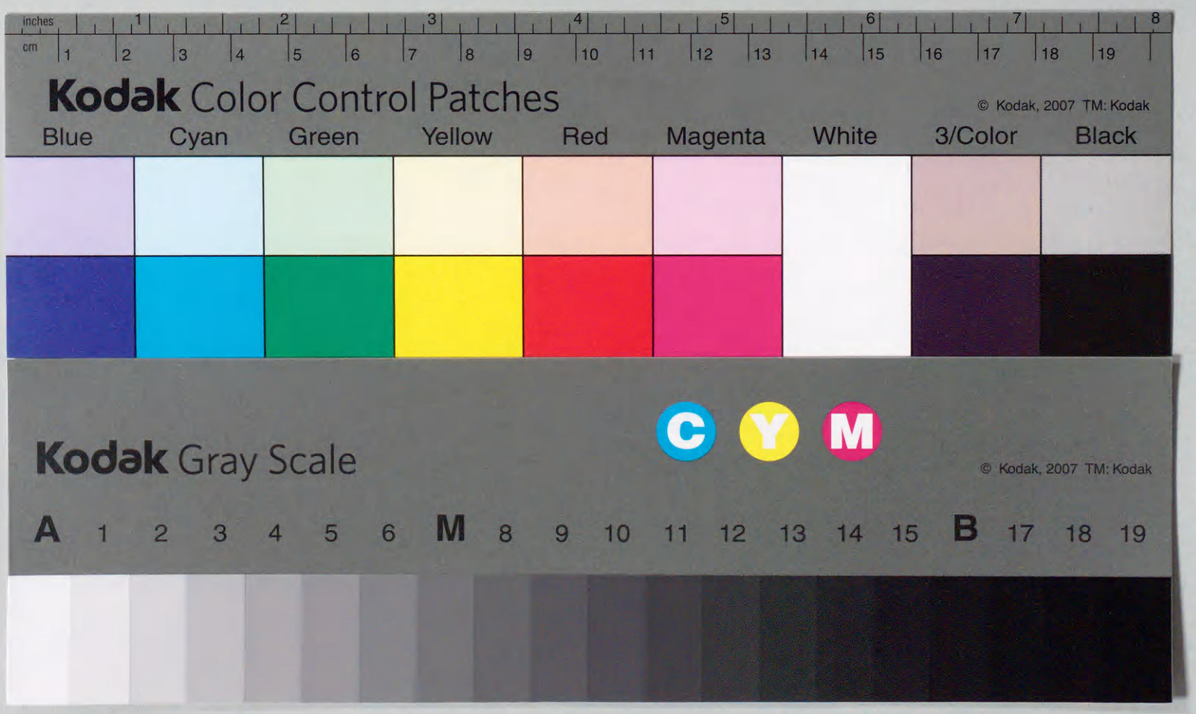
自然発話のための音声認識システムに関する研究

1996年1月

博士(工学)

甲斐充彦

豊橋技術科学大学



自然発話のための音声認識システムに関する研究

1996年1月

博士（工学）

甲斐　充彦

豊橋技術科学大学

自然発話のための音声認識システムに関する研究

論文要旨

音声認識の研究では、近年の計算機の処理能力の向上と、大量のデータベースを用いた統計的な手法の導入により大きな進展を見せた。これらの多くは、従来の制約であった孤立単語認識から連続音声認識へ、特定話者から不特定話者の認識へ、小・中語彙から大語彙へ、と目標を高くすることを可能にした。しかし、従来の音声認識技術はていねいに朗読された音声を対象とし、いわゆる書き言葉用の文法に従う発話を仮定してきたため、そのような制約から外れた発話に対しては十分な認識性能を保持することができなかった。また、自然な発話では、発音があいまいな冗長語や、言い直し、言い淀み、助詞落ちなどの現象が見られるため、従来のように音素や単語レベルの知識と、より上位の構文、意味的な知識を分割したアーキテクチャでは、認識性能の限界が予想された。

本論文は、こうした背景で、まず文脈自由文法による構文知識を音声照合の処理に採り入れ、統合的な音声の仮説照合と探索を行なうアルゴリズムを提案した。一つは、従来の単語レベルの照合を行なうワードスポッティング法の利用による拡張連続 DP 法の原理に基づいて実現し、もう一つはパターンマッチング問題としての連続音声認識において最適な探索を行なう One Pass DP 法の原理に基づいて実現した。前者は音声照合の計算量が語彙数のオーダである効率的な手法で、後者はビームサーチ法を用いて構文制約を動的に展開し、照合仮説の枝刈りを行なうことで計算量の増加を抑えた。実験的な検討に基づいて、それぞれのアルゴリズムの有効性を示した。

また、本研究では未知語・不要語を扱うための手法を検討した。一般に連続音声認識ではサブワード単位の音響モデルを用いることが多いため、未知語も何らかのサブワードの系列で表される。そこで、未知語としてのモデルを、任意のサブワードモデルの接続に対応させることで、既知語と未知語のそれぞれの尤度の比に基づく未知語の検出が考えられる。初めに、未知語・冗長語をそのような方法で処理することを試みた。このような方法は登録語以外の発話一般に対して適用できるので、発話のリジェクションとしての有効性も考えられる。このような手法の有効性を客観的に知るため、孤立単語認識のシミュレーション実験によって単語認識性能と未知語検出性能との関係を求め、実音声による未知語検出の実験においても同様な傾向があることを示した。

音声認識処理において間投詞、言い直しなどを扱うためには、一般に不要語としての処理が必要であり、倒置、助詞落ち、非文法的な発話などに関しては言語解析に関しての対処が必要である。これまで、自然な発話においてどのような認識手法が有効で

あるかの明確な比較はなされていない。そこで本研究では、不要語としての照合方式や言語解析法が異なる複数のシステムを実現し、自然な発話に対する認識実験によって比較・評価した。そして未知語検出などのために検討したサブワードモデルに基づく未知語処理法が自然な発話の認識において有効に働くことを明らかにし、さらに構文・意味的レベルの制約が統合された認識手法の有効性を示した。

A Study on Speech Recognition System for Spontaneous Speech

ABSTRACT

Recent progress in studies on speech recognition system has been mainly achieved by the improvement of the computational performance and by introducing the statistical methods into acoustical and language modeling with a large database. Such progress has made important contributions to relax the restrictions of speech recognizers: from isolated word recognition to continuous speech recognition, from speaker dependent recognition to speaker independent recognition, and from small/medium vocabulary recognition to large vocabulary recognition. However, such methods could not be directly applied to spontaneous speech since they had usually assumed to account for read speech and that the utterance followed a grammar for written language. In spontaneous speech, the limitation of the performance is expected in conventional methods based on a hierarchical architecture since spontaneous speech involves ambiguous pronunciations as well as the insertions of interjections, restarts, hesitations and ellipses of postpositional particles (disfluencies and ill-formed constructs).

This thesis discusses an attempt to dealing with spontaneous speech on the basis of using a statistical acoustic model and pattern matching techniques. First, two algorithms which integrate the speech verification and syntactic analysis are presented. The algorithms incorporate the syntactic constraints into speech verification process to find out an optimal or sub-optimal solution with respect to the pattern matching problem. Both of the methods assume that the syntactic knowledge is represented by a context-free grammar which is suitable for describing natural language. One of the methods employs word spotting methods, which have also been used in our conventional system, based on the augmented continuous DP method. Another method is proposed on the basis of an optimal pattern-matching based algorithm, the One Pass DP method. The former approach achieves an efficient process which only needs to perform the word verification in the order of vocabulary size. The latter approach attempts to find out an optimal solution with a little increase in computation with respect to the former approach by employing the beam search method to derive a dynamic constraint for search space and a pruning method to avoid the verification for unlikely hypotheses.

This thesis also investigates the method for dealing with unknown or the out-of-vocabulary words in continuous speech recognition. Such a method is necessary also for dealing with spontaneous speech since the extraneous speech such as interjections and restarts can be processed in the same way. In general, the out-of-vocabulary words can be represented by an arbitrary sequence of subword units if the vocabulary word is constructed by the concatenation of acoustic model based on a subword unit. Thus, the out-of-vocabulary words are detected with respect to the likelihood ratio between the hypotheses which correspond to the registered word and out-of-vocabulary word, respectively. The approach is effectively applied to our speech recognition system. Some experimental results show the effectiveness of this approach using the test utterances which include out-of-vocabulary words and interjections. This approach can also be applied to the rejection of a sentence hypothesis of the recognizer output. To know objectively the effectiveness of this rejection method, some experiments by simulation of isolated word recognition are carried out and the relationship between the word recognition accuracy and the correct rejection rate is reported.

It is necessary for speech recognition system to identify the extraneous speech such as interjections and restarts for dealing with spontaneous speech, as well as to parse illformed sentences which involve the inversion, ellipses of postpositional particles and ungrammatical sentences. Although the spontaneous speech has the acoustic and linguistic phenomena of a different nature, the explicit comparison of the methods for speech and language processing has not been performed. Thus, this thesis attempts to compare different search and parsing strategies by realizing the different experimental systems along with a proposed One Pass method-based system which includes unknown word processing.

目 次

図目次

表目次

第1章

序論

近年では、コンピュータで音声や画像をはじめとした複数の情報通信の媒体を扱うことが容易になり、ヒューマン・マシンインタフェース技術の果たす役割が重要になってきている。その中で、音声による人と機械とのインタフェースに関しては、古くから実現が期待されている技術の一つである。

音声認識の技術は、過去約20年余りの間に大きく発展してきている。しかしながら、音声認識が現在でもユーザインタフェースとして広く利用されていないのは、現在の技術においても残されている様々な問題や制約があるからである。次節ではそのような問題点について概観し、それに対して本研究が取り扱う問題や目的などについて1.2節で述べる。最後に、1.3節で本論文の構成について述べる。

1.1 研究の背景

音声認識の研究では、最近の音声認識技術の発展とコンピュータの処理能力の著しい向上によって多くの成果が得られた。それらの多くの成果は、これまでの音声認識の研究での中心的なテーマが (1) 孤立単語音声の認識から連続音声の認識へ、(2) 特定話者から不特定話者の認識へ、(3) 小、中語彙から大語彙へ、と移っていったことに象徴されている。このような研究の流れの中では、おもに幾つかの研究プロジェクトが先導的な役割を果たしてきた。米国では、1970 年から 5ヶ年計画で DARPA(Difence Advanced Research Projects Agency) による音声理解システムの研究プロジェクトが始められた[1]。その頃から、それまでのディクテーションが中心だった音声認識技術に対して、発話内容の意味的な理解を意識した音声理解の技術開発が活発になった。米国では、更に 1985 年から DARPA の「不特定話者大語彙連続音声認識」の研究が開始された[2, 5]。ヨーロッパでも、英国の Alvey プログラム、フランスの GRECO 計画、EC 諸国間の ESPRIT プロジェクト、NATO を支援組織とする RSG10 など、国家的あ

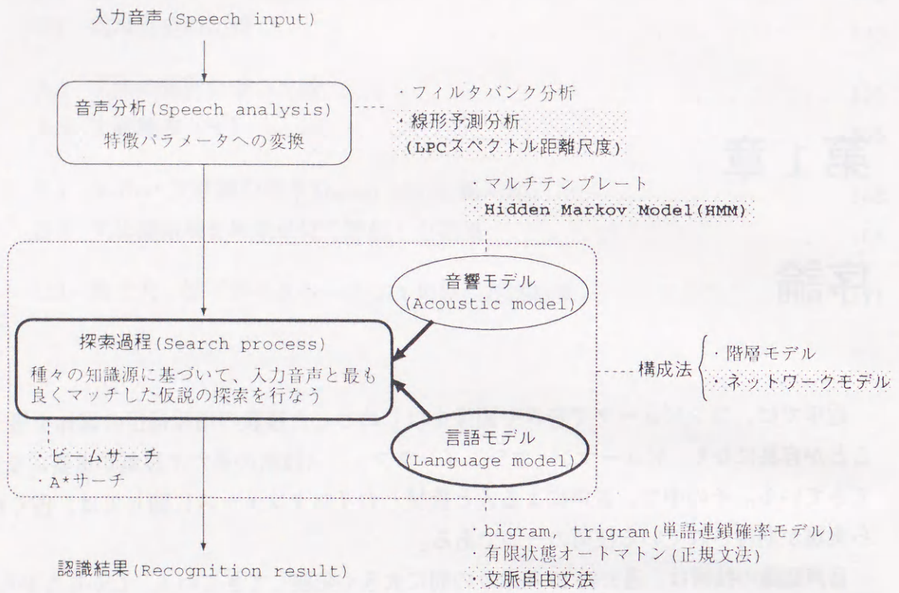


図 1.1: 音声認識システムの一般的な構成と要素技術
(太字・網掛けは本論文で採用したモデル・手法)

るいは国際的な規模でヒューマン・マシンインターフェースとして音声入出力の研究が行なわれている[3, 4]。

音声認識研究では上述のようなプロジェクトにより、LPC スペクトル距離尺度、DP マッチング、Hidden Markov Model(HMM)、ビームサーチ、n-gram など多くの要素技術とシステム化の成果があった。図 1.1は、2 種類の知識源のモデルを仮定した音声認識システムの一般的な構成とそれらの要素技術との対応を示している。音声認識システムは、複数の知識源による探索過程の制御の実現方式によって、主に階層モデルやネットワークモデルなどで代表されるシステム構成が採られている。一般に、階層モデルの構成法は各処理レベル間の制約が少なく、独立した評価が行ない易いため、各処理レベルでの計算コストが比較的大きい知識源のモデルの利用が容易である。従って、例えば文脈自由文法や意味知識を含めた言語モデルは bigram や正規文法に比べて計算コストが大きくなるので、この構成法での適用例が多い[11, 12]。一方、ネットワークモデルの構成法は、より大域的に最適な結果が得られ易い長所があるが、処理量が一般に多くなり、各知識源による制約を一つの制御構造にまとめることが困難であるため、言語モデルとしては bigram や正規文法での実装が多い[14]。最近では、N-best

パラダイムのように段階的に制約の異なる知識源を用いる探索戦略が多く[16, 18]、そのようなシステムでは上述のような構成法が更に組み合わされたものとなっている。

従来、このような要素技術の多くは、読み上げ音声（朗読音声, read speech）を認識の対象として研究が行われてきた。しかし、近年になって自然な発話 (spontaneous speech) を対象とした研究が活発に行われるようになっている。DARPA の研究プロジェクトも、1988 年から 1989 年にかけて、従来の不特定話者の大語彙連続音声認識の研究から、「音声言語システム (Spoken Language System)」と呼ばれる音声言語インターフェースの研究に移行し、音声のより言語的な側面にアプローチすることが意識され始めている[6]。

著者の研究室では、音声対話システムのフロントエンドとしての音声認識システムが開発されてきた。このシステムは、文脈自由文法の言語モデルを用いており、単語レベルの音声照合処理と、構文・意味レベルの言語処理からなる階層モデルの構成を採っていた。本システムにおいても、従来は読み上げ音声のみを対象として各処理レベルでの改良を行なっているため、より自然な発話に対して十分な性能が得られない問題があった。その主な要因として、まず音響的な特徴の違いが挙げられる。例えば、音素レベルでの発音も自然な発話ではあいまいになりがちであり、音素認識精度の低下が報告されている[67, 68]。また、言語的な特徴の違いも大きく、従来の書き言葉対象の文法では言語制約の範囲外となるような自然な発話特有の言語現象による問題が挙げられる。代表的なものとして、間投詞、助詞落ち、倒置、言い直し、言い淀み、未知語の出現がある。これらの特徴の違いが認識性能に与える影響は一般に極めて大きいといえる。そこで、自然な発話に対して頑健 (robust) な音声認識システムについて検討することが必要であり、そのようなシステムを構築するために以下のような課題について考えることが重要といえる。

1. 音声処理と言語処理を統合した認識手法

   従来の多くのシステムは、音声処理と言語処理が密に結合していない階層型の構成をとっているため、モジュール間で渡される情報が十分でなく、認識精度の劣化が指摘されている。特に、自然な発話では言い直しや間投詞などの音声現象が含まれ、一般的な音響モデルの基本単位である音素や音節レベルでのあいまいさは更に増加することが予想されるので、言語知識を含めた統合的で効率的な認識手法が重要になるものと考えられる。

2. 未知語を扱うための認識手法

   従来の音声認識システムでは、未知語 (out-of-vocabulary words) に関して特別な処理をしていないため、未知語が用いられる状況では信頼できる認識結果が得ら

れないという問題がある。そこで、未知語をより確実に検出・認識する手法が望まれる。

3. 信頼性の低い認識結果の棄却

音声認識システムでは、信頼性が低い認識結果に対して、より確実に棄却するような機能の実現が望まれる。これは、未知語を含む発話や文法外(out of grammar)の発話などを扱う問題とも関係しているといえる。また、対話システムの場合、対話を通して認識結果の信頼性が低い部分を回復または聞き直しできる可能性があるので、特に有用と思われる。

4. 自然な発話に対して頑健な認識手法と言語処理

対話の中での自然な発話(spontaneous speech)には、音声言語に特徴的な、間投詞、言い直し、倒置や助詞落ちなどの現象が含まれる[67, 69, 71, 72]。また、対話中の表現は、書き言葉の文に比べて断片的で不完全なものになりやすい。それらの現象に対処するための認識手法については近年研究が行われ始めたところであり、従来の認識手法も含めてどのような方法が自然な発話の認識・理解のために有効であるかを検討する必要がある。

上記1番目の課題に対して、文脈自由文法[1]による言語的制約の下での統合化を行っている例として、CYK 法に基づく方法[44]、LR パーザに基づく方法[28, 29]、チャートパーザに基づく方法[45]などが提案されている。CYK 法に基づく方法は bottom-up 型の解析アルゴリズムなので、一般に予測的な情報を音声認識のレベルに十分反映できない問題がある。LR パーザに基づく方法は、あらかじめ文法規則から LR 解析表に変換する必要はあるが、LR 解析表を参照しながら LR パーザを予測的に用いることによって効率的な言語処理を実現することが可能である。通常の LR 構文解析法では曖昧性を持った文脈自由文法を扱えないが、拡張された一般化 LR 構文解析法(generalized LR parsing)によって扱うことが可能となる[28]。チャートパーザによる方法は、解析木のグラフ表現であるチャートを生成していく過程によって解析を行なうもので、トップダウンによる解析法を予測的に用いて One Pass DP 法の構文制御に用いる方法が提案されている[45]。また、上述のように句構造文法を用いない方法も考えられ、例えば、句レベルを再帰的な遷移ネットワークで表現し、句間の接続は統計的言語モデルで表現するような言語制約の統合による方法もある[31]。

上記2番目の課題に対しては、ある認識タスクの実現に必要なキーワードだけを考慮し、少数のキーワードのみを抽出するようなワードスポッティングベースの方式が検

[1]文脈自由文法は自然言語の多くの言語現象を表現でき、効率的な解析法が開発されているので、自然言語処理でよく用いられる。

討されている[57, 58, 88]。この方式では非キーワード(non-keywords)の過剰検出を抑えることが重要なので、非キーワードに対する音響的なモデル化が検討されている。一般的にはキーワード以外(non-keywords)の音声を少数の音響的モデル(filler モデル)で表現する方法が用いられる[55]。また、未知語や非キーワードのモデル化の方法として、サブワードモデルに基づく方法がある[57, 56]。この方法は、処理量は増加するが、ワードスポッティングベースの方式では garbage モデルに対して検出性能が一般に優れていることが示されており[57, 58]、最近では連続音声認識においても適用されている[56, 63, 64]。サブワードモデルに基づく方法では、未知語を明示的にモデル化する方法と[56]、音韻タイプライターと同様な処理を並行して行なう方法[63, 64]があり、後者では未知語の音素系列の推定が可能になる。

上記3番目の課題に対しては、キーワードスポッティング方式では、上述の非キーワードの過剰検出を抑える方法と同様に、filler モデル(garbage モデル)をキーワードの照合と並行して用いて、キーワードの尤度と非キーワードの尤度の尤度比に基づいて検出誤りを抑えることができる[57, 58]。同様に、連続音声認識でも、サブワードモデルに基づいて発話の棄却規準となる尤度を求めて発話のリジェクション(rejection; 棄却)を行なう方法の有効性が示されている[66]。

上記4番目の課題に対しては、現時点では従来の枠組においてわずかに改良して対処しているものが多い。自然な発話に特徴的な音声現象と言語現象に対処するためにそれぞれ検討されているアプローチは、主に以下のようなものである。音声現象に対しては、従来のようにワードスポッティングベースの方式によって不必要な単語(非キーワード、間投詞など)に対処するものや[58, 59]、キーワード抽出のタスクでこれまで検討されているように garbage モデルを用いてフレーズ間に挿入される間投詞や不必要な音声に対処する方法などがある[84, 65]。また、言語現象に対しては、単語や句単位のスポッティング又は統計的な言語モデルを用いて、単語や文節又は単語列の認識候補を求め、意味的な単位でまとめられた文法に基づいて部分的な解析(partial parsing)を行なうアプローチや[84, 85, 86]、文中の句に対応する概念の並びを Markov モデルに基づく確率的なモデルで表現して音声処理と統合する方法[87]、などが検討されている。これらの多くは、DARPA のプロジェクトの ATIS(Air Travel Information System)という明確な領域を対象として検討されている。

1.2 本研究の目的

本研究では、前の節で示した4つの課題に対して、次のようなアプローチで検討を行う。

1. Earley 法に基づく文脈自由文法の構文解析処理と音声（照合）処理の統合化

   Earley 法はトップダウン・横型探索に基づく構文解析法であるので、音声処理を時間同期的に進める処理との統合を検討する。この方法は、チャートパーザをトップダウンに適用する場合と同等である[45]。連続音声認識は一般に短い音響的単位に基づく言語的制約の下でのパターンマッチング問題であるが、文脈自由文法に基づくトップダウンの予測を用いる場合には、膨大な部分文仮説が生成され得るため枝刈りが必要である。そこで、本研究で検討した方法は、音声処理との統合によって可能となるフレーム単位での枝刈りやビームサーチ法によるトップダウンの予測を行い、音声処理のための探索空間を効率的に削減できる方法を実現する。また、音声処理との統合では、構文解析での文法的なあいまいさだけでなく、音声照合によるあいまいさのために複数の解析状態を並行して処理する必要も生じるので、解析処理の効率化と解析のあいまいさによる音声照合の重複を改善する方法を示す。

2. サブワード単位の音響的モデルに基づく未知語処理

   未知語はサブワード (subword) 単位（例えば音節）の連結で表現できるので、未知語のモデルをそのように仮定することで未知語仮説としての照合を行う方法を検討する。既に提案されている方法[56]と同様であるが、厳密な未知語処理を行なう場合の計算量は、生成される未知語仮説の数に依存して増加するので、大語彙の場合には処理量の増大が問題である。本研究では、計算量を削減する近似的な処理のためのアルゴリズムを示す。また、自然な発話では多くの種類の間投詞が観測されるので、間投詞を未知語として処理することを検討する。

3. サブワード単位の音響的モデルによる発話のリジェクション性能の評価

   未知語処理の考え方を発話全体に適用することによって、ある発話に対する認識結果の信頼性が低い場合にリジェクションを行う方法を検討する。これまでに同様な方法の有効性が示されているが、実際の音声認識システムの性能とリジェクション性能との関係が明らかではないので、この方法でのリジェクション性能と音声認識性能との関係について、シミュレーションにより検討する。また、孤立単語認識実験によってリジェクション性能を調べ、シミュレーション結果との相関を確かめる。

4. 自然な発話に対する音声認識処理での有効な照合・解析法の検討

   不要語に対処するために、一般にワードスポッティングベースの認識手法を採用する場合が多い。しかし、連続音声認識では、一般に認識スコアは仮説に対する

入力音声の尤度を与えるため、異なる区間の認識結果による認識スコアの比較は難しい。そこで、本研究で検討している未知語処理を適用して認識スコアの算出を改善したり、言語処理を統合した One Pass アルゴリズムにおいて未知語処理を導入した場合について有効性の比較・検討を行う。また、入力文に対する頑健な解析を実現するために、意味的な制約に基づく倒置を許容した文節単位の文法の利用を考える。構文の制約が緩い場合には、従来のように left-to-right に解析する方法においてもかなり探索空間は増大することが予想されるため、尤もらしい単語から優先して解析する island-driven 法（島駆動方式）との比較も試みる。

本研究の実験に用いた音声認識システム SPOJUS-SYNO(SPOken Japanese Understanding System - SYNtax Oriented) の、これまでの開発の経緯を表 1.1に示した。バージョン I 及び II は従来のシステムである[22]。本研究では、前述の 1 番目の検討においてバージョン III（3章）、X のシステム（3章）を実現し、2 番目の検討における未知語処理法をバージョン X のシステムに対して追加・拡張してバージョン Y のシステム（4章）を実現した。話者適応化法については本研究では特に触れない[75]。

表 1.1: 音声理解システム SPOJUS-SYNO の開発の経緯

Version	特徴
I	• 音節単位連続出力分布型 HMM (全共分散行列、離散分布型継続時間制御) • ワードスポッティング （音声特徴パラメータ：メルケプストラム＋回帰係数） • ワードラティスのフレーム同期型構文解析 （Earley 法による文脈自由文法制御、ビームサーチ法）
II	＋ 話者適応化（最尤推定学習）
III	ワードスポッティングと構文解析のフレーム同期処理による統合化 （拡張連続 DP 法の導入）
X	フレーム同期型 One Pass アルゴリズム （Earley 法に基づく文脈自由文法制御の統合）
Y	＋ 間投詞、言い直し、未知語処理 ＋ オンライン話者適応化（最大事後確率推定法）

1.3 本論文の構成

2章では、音声認識システムを構成する要素技術および構成法などについて概観する。また、音声認識システムの認識性能の比較において重要な、言語的な複雑性の尺度について述べる。言語的な複雑性の尺度は、その制約の強さが音声認識の困難さとおおよそ相関があるような尺度であり、異なる言語知識を用いたシステム間の性能の比較を行う場合の規準として一般によく用いられるものである。

3章では、構文の制約を用いた連続音声認識システムにおいて、音声処理と言語処理の統合を実現するための検討について述べる。始めに、音声処理部と構文解析部が分かれた従来の階層型のシステムの概略を説明し、言語処理を音声処理に統合するための2種類のアプローチを示す。これらはEarley法に基づくtop-down型構文解析法に基づいており、その効率的な適用法とフレーム同期型の音声認識アルゴリズムとしての統合の方法について述べる。続いて、これらの統合化したシステムにおける認識実験を通して、2種類のアプローチで実現したシステムの比較と、種々の検討による処理の高速化の効果について示す。

4章では、未知語や不要語を扱うためのサブワードモデルを用いた未知語処理法の実現と実験的な評価について述べる。まず不要語の典型である間投詞に関して、音声対話データベースの分析結果を示す。次に、未知語処理のためのアルゴリズムについて一般的な解法と近似的な解法を与え、認識実験による未知語処理の有効性を示す。更に間投詞を未知語処理によって処理した場合についても間投詞を登録する方法と比較し、未知語処理の有効性を示す。

5章では、未知語処理と同様な考え方による発話のリジェクションの性能について評価する。まず、シミュレーション法により音声認識システムを近似し、認識性能と未知語のリジェクション性能との関係を示す。更に、実際の音声による孤立単語音声認識によって同様なリジェクション性能を求め、シミュレーションとの関係を調べる。

6章では、いくつかの照合・解析法に基づいて、自然な発話に対して有効なシステムの認識アルゴリズムの構成を検討する。そして、評価実験によってそれらの方法の優位さの比較を行う。

最後に、7章で結論と今後の課題について述べる。

第2章

音声認識概論

2.1 はじめに

音声の自動認識が困難なのは、音声パターンのあいまい性にある。あいまい性には1) 個人差、2) 調音結合、3) 発声速度等に起因するものがある。さらに連続音声の認識においては1) 単語境界が不明確になり、2) 単語間で調音結合が起こり、3) 不正確な発声になりがちであり困難度は増大する。

これらのあいまい性に対処する方法としてDPマッチング、マルチテンプレート法、HMMによる確率的処理などが開発されている。また文音声の認識処理では、構文・意味・文脈などの言語的知識に基づいて入力音声の仮説をたて、その仮説の妥当性を音響分析・音素認識・単語認識などの結果を用いて検証し、妥当性のスコアが最大になる仮説文を認識出力する方法が一般的である。仮説の生成・検証の処理過程は膨大な候補の中からの最適値探索問題である。この章では、まず音声認識システムを構成する音声処理と言語処理の基本技術について概観し、更に連続音声認識システムの評価において認識の困難さを測る目安として用いるタスクの言語的な複雑性の尺度について述べる。

2.2 音声認識処理

2.2.1 音声の前処理[39, 52, 7]

音声の前処理は、音声の分析による特徴量の抽出と、観測された入力信号波形からの音声区間の検出である。以下に、各部分での処理の概要について述べる。

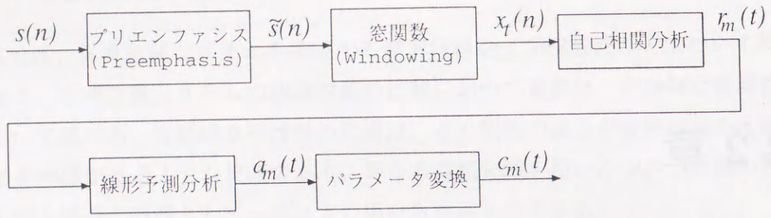


図 2.1: 音声分析の過程

(a) 音声の分析

音声信号の分析の方法としては、短時間パワー (short time energy)、零交差数のような比較的簡単に抽出できるものがあるが、音声のパラメータ表現として最も一般的なのは短時間スペクトル (short time spectral envelope) である。スペクトル分析方法には、音声の生成モデルを仮定したパラメトリック分析法と、仮定しないノンパラメトリック分析法がある。ノンパラメトリック分析法として、フィルタバンクを用いた分析法や高速フーリエ変換 (FFT)、及び FFT ケプストラム法などがあり、パラメトリック分析法として線形予測分析法、LPC ケプストラム法 (LPC: Linear Predictive Coefficient、線形予測係数) などがある。以下では、本研究の音声認識システムの音声分析処理として採用している、線形予測分析に基づく音声分析の処理過程の概要について述べる。

音声分析は図 2.1の手順で行なわれる。プリエンファシス (preemphasis) の処理は、一般に低い次数のディジタルフィルタ（一般に 1 次 FIR フィルタ）であり、スペクトルの平坦化（高域周波数帯域における S/N 比の向上）と有限桁数でのディジタル信号処理の演算誤差をなるべく抑えるために用いられる。最も良く用いられるのは、次のような 1 次のシステムである。

$$H(z) = 1 - a\,z^{-1}, \qquad 0.9 \le a \le 1.0 \tag{2.1}$$

音声信号は非定常確率過程であるが、短時間単位では定常確率過程とみなして分析される。音声を分析する際には分析窓を用い、音声波形を一定の周期毎（通常は 10ms 単位くらい）に切り出す[7]。窓長 Tは、短いとスペクトルの時間分解能は向上するが周波数分解能が劣化し、長いと周波数分解能は向上するが時間分解能が劣化する。後述の自己相関分析においては、分析の対象となる音声セグメント $x_t(n), 0 \le n \le N-1$ の区間外の信号が零であると仮定する必要があるが、切り出しによるスペクトルへの悪影響を抑えるために、一般に切り出しの始まりと終わりの部分が零に近いような窓関数（例えば、Hamming 窓）を用いる。窓関数を用いた場合、処理時刻 t の音声セグメント $x_t(n)$ は次式で求まる。

$$x_t(n) = \tilde{s}(t+n)w(n), \quad 0 \le n \le N-1 \tag{2.2}$$

但し、$w(n)$ は窓関数で、本研究では次式で表される Hamming 窓を用いる。

$$w(n) \;=\; 0.54 - 0.46cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N-1 \tag{2.3}$$

図 2.1の自己相関分析は、次に述べる線形予測分析で用いられる。最後のパラメータ変換は、後に述べる LPC ケプストラムや LPC メルケプストラムへの変換である。

(b) 線形予測分析

線形予測分析法は、声道のモデルとして全極型伝達関数を仮定したもので、ある時点のサンプル値がそれ以前のいくつかのサンプル値の線形結合で表されるというモデルの仮定による分析方法である。すなわち、音声信号 $x(n)$ は過去の p サンプルによって次のように近似される。

$$x(n) \approx a_1 x(n-1) + a_2 x(n-2) + \cdots + a_p x(n-p) \tag{2.4}$$

この式を音源項を含めて等価に表すと、

$$x(n) = \sum_{i=1}^{p} a_i x(n-i) + Gu(n) \tag{2.5}$$

となり、伝達関数は、

$$H(Z) = \frac{X(Z)}{U(Z)} = \frac{G}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \frac{G}{A(z)} \tag{2.6}$$

と表される。ここで、過去のサンプルからの推定値、

$$\tilde{x}(n) = \sum_{i=1}^{p} a_i x(n-i) \tag{2.7}$$

を考えると、予測誤差 $e(n)$ は、

$$e(n) = x(n) - \tilde{x}(n) = x(n) - \sum_{i=1}^{p} a_i x(n-i) \tag{2.8}$$

となる。線形予測分析の基本的な問題は、式 (2.6) のディジタルフィルタの特性が分析窓の音声波形に一致するような予測係数 $\{a_k\}$ を求めることである。音声のスペクトル特性は時間と共に変化するので、ある短時間セグメント内の音声信号から予測係数を推定しなければならない。結局、線形予測分析の方法は、短時間セグメント内での 2 乗平均予測誤差 Eが最小になるような予測係数を求めることである。

$$E = \sum_n e^2(n) = \sum_n \{x(n) - \sum_{k=1}^{p} a_k x(n-k)\}^2 \tag{2.9}$$

$$\frac{\partial E}{\partial a_i} = \sum_{k=1}^{p} a_k \sum_n x(n-k)x(n-i) - \sum_n x(n)x(n-i) = 0 \tag{2.10}$$

$\{a_i\}$ を求めるには、$\sum$の範囲により自己相関法 ($-\infty < n < \infty$, 窓以外では $x(n) = 0$) と共分散法 ($0 \leq n \leq N-1$) があり、次の連立方程式を解いて求める。

・自己相関法

$$\sum_{k=1}^{p} a_k R(i-k) = R(i) \qquad (1 \leq i \leq p) \tag{2.11}$$

$$R(i) = R(-i) = \sum_{n=-\infty}^{n=\infty} x(n)x(n+i) \tag{2.12}$$

・共分散法

$$\sum_{k=1}^{p} a_k \varphi_{ki} = \varphi_{0i} \qquad (1 \leq i \leq p) \tag{2.13}$$

$$\varphi_{ik} = \varphi_{ki} = \sum_{n=0}^{N-1} x(n-i)x(n-k) \tag{2.14}$$

本研究では自己相関法を用いている。

(c) 特徴パラメータの変換

線形予測分析に基づくケプストラムは、LPC 対数スペクトル包絡の逆フーリエ変換で定義されるが、線形予測係数とケプストラムの関係から直接求めることができる。

$$c_0 = ln\sigma^2 \tag{2.15}$$

$$c_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, \quad 1 \leq m \leq p \tag{2.16}$$

$$c_m = \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) c_k a_{m-k}, \quad m > p \tag{2.17}$$

但し、σ^2は LPC モデルにおけるゲイン項である。一般にケプストラムの次元数 Q は $Q > p$ で、$Q \simeq \frac{3}{2}p$ である。

ケプストラム係数 $\{c_n\}$ は、更に周波数軸をメル尺度に非線形変換することでメルケプストラム係数に変換され、音声認識で用いられることが多い。人間の聴覚特性は、音の大きさに対してはほぼ対数的な特性で、周波数分解能は低い周波数では細かく、高い

周波数では粗いメル尺度で特徴づけられる。前者はケプストラムで既に表現されているので、周波数軸に関してこのような変換が用いられる。LPC メルケプストラム係数は、LPC ケプストラム係数 $\{c_n\}$ から、1 次の全域フィルタ $H_\alpha(Z) = (Z^{-1} - \alpha)/(1 - \alpha Z^{-1})$ の位相特性を用いた周波数変換で近似的に求められる[93, 94, 39]。

上述のケプストラム、メルケプストラム係数は、局所的なスペクトルの特徴をよく表現しているが、音声認識では更にスペクトルの動的な特徴を用いることの有効性が示されている。本研究では、メルケプストラム係数に加えて、各次元毎の回帰係数により新たに構成される特徴ベクトルをスペクトルの動的特徴量として用いる。時系列 $x(t)$ において、時間 t を中心とした幅 $2w+1$ の線形回帰係数$\Delta x(t)$ は次式で計算される。

$$\Delta x(t) = \frac{\sum_{i=-w}^{w} i \cdot x(t+i)}{\sum_{i=-w}^{w} i^2} \tag{2.18}$$

結果的に、メルケプストラム係数の次元数が Q のとき、回帰係数による動的特徴量を加えた特徴ベクトルは $2Q$ 次元のベクトルとして表される。

(d) 音声区間検出

音声認識システムにおいては、音声パターンのテンプレートの作成や音響的モデルの学習のために音声区間の検出が必要である。音声認識では音声区間の検出精度が認識精度に影響するため、様々なアプローチが検討されている。大きな分類に基づくと、そのようなアプローチは (1)explicit なアプローチ、(2)implicit なアプローチ、(3) ハイブリッドアプローチがある[52]。その特徴は、(1) は音声区間の検出を、その後のパターンマッチングの処理内容とは独立に行ない、(2) は音声区間の検出とパターンマッチング処理を同時に行ない、(3) はあらかじ幾つかの音声区間の候補を検出し、(2) と同様にパターンマッチング処理において最も良い区間を検出する、というものである。ノイズが比較的少ない環境では、(1) の方法でも検出の精度は良い。本研究では、主に静かな環境での録音音声を対象とするため、(1) の方法に基づいて、音声の一フレーム毎のパワーをあらかじめ定められた閾値と比較することによって音声区間を検出している。なお、最近、著者の研究室では雑音モデルを用いて音声区間の始端を検出する方法を用いている。

2.2.2 音声の照合

(a) DP マッチング法

音声認識は基本的に、入力パターンと標準パターン（テンプレート）とマッチングを行い、発声がなんであるか識別する処理である。パターンが固定長であれば簡単で

あるが、可変長である場合は両者の長さを整合させねばならない。代表的で強力な手法は動的計画法（Dynamic Programming :DP）を用いる DP マッチング法である。DP マッチング法は時間軸非線形伸縮パターン整合法であり、発声速度の変動を吸収する。話者変動には標準パターンの複数化（マルチテンプレート法）で対処可能である。

(b) HMM 法

HMM(Hidden Markov Model: 隠れマルコフモデル) は、入力音声を時間的に小さな構成単位（シンボル）の系列として表現し、そのシンボル系列が隠れマルコフモデルによって生成されたものと仮定し、その生成確率を最大にするモデルを認識結果とするものである。音声の細部構造を考慮して音韻を記述することができ、音声のスペクトル的な揺らぎも時間伸縮の揺らぎも、実際の音声のゆらぎを統計的にモデル化するため特に発声の変動が大きい不特定話者音声認識や連続音声認識に適している。2.4.2 節で詳述する。

(c) ワードスポッティング法

音声中から特定の単語を検出する方法である。連続音声では単語境界があいまいであるため、異なった位置の複数個の単語候補を仮定せざるをえない。また文のキーワードとなる単語を検出する場合にも用いられる。実際のインプリメンテーションは、連続的な DP マッチング法の考えを基本にしている。3.4.2節で詳述する。

2.3 言語処理

連続音声の認識においては具体的なタスクを設定することにより、単語音声の場合にはなかった種々の情報を利用することができる。言語処理は一般的に、単語レベル、構文レベル、意味レベルの処理に分けられ、それぞれのレベルで辞書や構文規則などの知識を利用した解析を行なう。文法、意味、文脈などの様々な高次知識を用いることができるので、音響的特徴のみによる認識結果を補い、最終的な認識制度を向上させうる可能性がある。タスクを規定する代表的な言語モデルとしては、以下のものがある。

(a) bigram、trigram

単語・品詞の 2,3 字組確率を用いて単語列（文）の生起確率を近似するものである。2,3 字組確率は訓練サンプルから統計的に求められるが、大量の訓練サンプルを必要

とし、タスクで扱う語彙数が多い場合には文の生起確率を精度よく近似することは難しい。

(b) 文脈自由文法

文脈自由文法はもともと文を生成するモデルとして作られたものであるが、これを用いて与えられた文を解析して文の構造を明らかにし、タスクで規定された文のみを受理することができる。文脈自由文法は自然言語文法のモデルとして適当であること、そして効率のよい構文解析法が知られていることから、文認識システムの言語処理部でよく使用される。音声認識で用いられる構文解析法には bottom-up 的処理の CYK(Cook-Younger-Kasami) 法、top-down 的処理の Earley 法、決定的解析法である LR 法、などがある。なお、あいまいな文脈自由文法も扱えるように拡張された一般化 LR 法[8, 28]も最近よく用いられている。本論文では Earley 法に基づく方法を用いている。3.3.2節で詳述する。

(c) ATNG(Augmented Transition Network Grammar)

文脈自由文法をネットワーク表現したもの（基本遷移ネットワーク）の各枝に、i) 種々のレジスタに値をセットする機能、ii) 木構造の一部を記憶しておく機能、iii) 各種の検査を行なう機能を付加したものを拡張遷移網文法（ATNG）と呼ぶ。自然言語のよいモデルとされている変形文法と等価な機能を持つ。

(d) 係り受け（依存文法）

英語のような語順に強い規制がある言語は、文脈自由文法がよい近似を与える。日本語の場合は、語順が比較的自由であるものの文節の構造は安定であることから文節構造は正規文法で表現でき、文節間構造は文節間の依存関係（係り受け）で規定する方法がある。通常は係り受け関係では構文上の制約が緩いため格文法や結合価文法を併用する場合が多い。具体的には連続音声中から文節候補をスポッティングし、それらの連結を係り受けで制御する。

2.4 連続音声認識システムの構成

2.4.1 構成モデル及び処理方式[9, 10]

(a) 構成モデル

音声認識・理解系のシステムでは仮説をたて検証する手法が基本になっている。知識を利用した各処理レベルの仮説・検証をどのように制御するかによってつぎのような基本的なモデルに分けられる。

(i) 階層モデル

関連する相互の処理レベル間でのみ制御のやりとりを行い、全体の処理を階層化した形態である。簡潔で理解しやすく、多くのシステムで用いられている。処理の流れはトップダウン的なものと、ボトムアップ的なものとがある。しかし各種の知識が相互に関連しているため、単純な階層モデルのみでは、システムの性能に限界がある。代表的なシステムに BBN 社の HWIM[11]、京都大学の LITHAN[12] などがある。本研究の出発点となったシステムは、この階層モデルに属するシステムであった。

(ii) ブラックボード モデル

「黒板」と呼ばれる共通のデータベースを利用したもので、各々の処理レベルはこの黒板を介して通信しながら独立に処理を進める。知識処理技術として使われているブラックボードモデルは、CMU の音声理解システム Hearsay-II[13] がその始まりである。

(iii) ネットワークモデル

全ての処理レベルの知識を一つのネットワークに組み込み、一様な制御機構によって統一的に扱うものである。CMU の Harpy システム[14] がこの形態である。本研究で開発したシステムはネットワークモデルに該当するシステム構成になっている。

(b) 処理の順序

入力音声に対して、仮説・検証の処理を進める方法として left-to-right 法と island-driven 法とがある。

(i) left-to-right 法

時間軸に沿って順序よく処理する方法であり、制御は簡単である。しかし発話の先頭から順にその内容を確定できるという保証はなく、何らかの形でバックトラックの機構が必要になる。日本語の場合、述語を先に同定することで構文の制約が利用できるため、right-to-left に処理を進めることがある。Hearsay-I、LITHAN、Harpy などが採用している。本研究も left-to-right 法を用いている。

(ii) island-driven 法

信頼度の高い部分（島）から処理を始める方法であり、制御機構は複雑である。文脈が不明確なため限定作用が弱くなり、考慮すべき仮説が増え、多くの処理時間を要するが性能向上が期待できる。HWIM システムで採用されている。本研究では island-driven 法と left-to-right 法との比較を行なう。

(c) 木探索方式

処理レベルにおいて、いくつかの仮説を順に評価して確からしい解を探索する方式として次のような木（仮説）探索方式がある。音声認識では更に、仮説の深さが、入力音声の処理フレームの長さに従う方式と、音素や単語などの数に従う方式がある。

(i) best-first 法

評価値の高いものを択一的に選択しながら処理していく方式。評価関数の設定如何によって良否が決まる。未知同定区間の推定評価値を実際の推定値よりよい値を採用すれば最適解が得られる保証があり、A*探索法と呼ばれ最近音声認識でも良く用いられるようになった[15, 16, 17]。

(ii) depth-first 法

最も長いパスから順に処理していく方式。同じ長さの場合には評価値の高いものから処理していく。

(iii) breadth-first 法

パスの短いものから順に、同じ長さのパスをすべて並行して処理する方式。

(iv) beam-search 法

同じ長さのパスのうち評価値の高いものをいくつか並行して処理する方式。breadth-first 法に枝刈りを採用したもの[12, 14]。本研究も beam-search 法を用いている。

2.4.2 HMM を用いた音声認識[39]

音声認識で用いられる HMM は、left-to-right モデルと呼ばれ、初期状態と最終状態が設定されて、1 回の状態遷移毎にラベルを 1 つずつ出力する。HMM は、出力ラベルによって一意に状態遷移先が決まらないという意味での非決定性有限状態オートマトンとして定義される。次にどの状態に遷移するか、またその際にどのラベルを出力するかは、それぞれ、「遷移確率」、「出力確率」によって統計的に決められている。通常のマルコフモデルと異なる点は、出力ラベル系列が与えられてもその状態系列は唯一に決まらない、観測できるのはラベル系列だけであり、状態そのものは直接観測できない点にある。その意味で“hidden”マルコフモデルと呼ばれる。多種の出力ラベルが存在する場合、各々のラベルの出力確率の集合を 1 つの出力確率分布としてとらえることができる。その出力確率分布は、離散的な場合（ベクトル量子化を使用する場合）と連続的な場合（ベクトル量子化を使用しない場合）の 2 つに大きく分けることができ、それぞれ離散出力確率分布 HMM、連続出力確率分布 HMM と呼ばれている。簡単な例を図 2.2に示す。図中の a_{ij} は状態 q_i から q_j へ遷移する確率で $b_{ij}(k)$ はそのアーク (arc) でラベル k を出力する出力確率である。これらの確率は、次の拘束条件を満足する。

$$\sum_j a_{ij} = 1, \quad \sum_k b_{ij}(k) = 1 \tag{2.19}$$

この場合、観測ラベル系列が abb であったならば、可能な状態遷移系列は $q_1q_1q_2q_3$ と $q_1q_2q_2q_3$ の 2 つとなる。ここで $\boldsymbol{o} = o_1o_2\cdots o_T$ と $\boldsymbol{x} = x_0x_1\cdots x_T$ をそれぞれ出力と確率変数の系列とすると、単純マルコフ過程と隣接フレーム（ラベル）間独立の仮定とから、

$$P(\boldsymbol{x}) = \prod_i P(x_i|\boldsymbol{x}_1^{i-1}) = \prod_i P(x_i|x_{i-1}) \tag{2.20}$$

$$P(\boldsymbol{o}|\boldsymbol{x}) = \prod_i P(o_i|\boldsymbol{x}_1^{i}) = \prod_i P(o_i|x_{i-1}, x_i) \tag{2.21}$$

により、

$$\begin{aligned} P(\boldsymbol{o}) &= \sum_{\boldsymbol{x}} P(\boldsymbol{o}|\boldsymbol{x})P(\boldsymbol{x}) \\ &= \sum_{\boldsymbol{x}} \prod_i P(x_i|x_{i-1})P(o_i|x_{i-1}, x_i) \end{aligned} \tag{2.22}$$

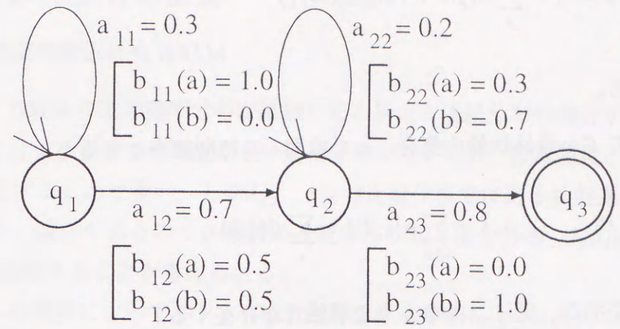


図 2.2: left-to-right 型 HMM の例

図 2.2の例において、式 (2.19) の右辺の和の中が 0 とならないのは、前述した 2 つの状態系列のみである。各々の状態系列での出力系列の生起確率を $P_1(abb), P_2(abb)$ とすると、

$$P_1(abb) = 0.3 \times 1.0 \times 0.7 \times 0.5 \times 0.8 \times 1.0 = 0.084 \tag{2.23}$$

$$P_2(abb) = 0.7 \times 0.5 \times 0.2 \times 0.7 \times 0.8 \times 1.0 = 0.0392 \tag{2.24}$$

となるから、$P(abb)$ は、

$$P(abb) = P_1(abb) + P_2(abb) = 0.084 + 0.0392 = 0.1232 \tag{2.25}$$

となる。問題は式 (2.22) をいかに効率よく計算するかである。一般にこの問題は、次に示す前向き確率のアルゴリズムにより効率的に解くことができる。

- 前向き確率と後向き確率

  ここで、$\alpha(i,t)$ を初期状態から始まり $o_1, o_2, \cdots, o_t$ を生成して状態 i に達する確率（前向き確率）とする。この前向き確率は、次のように再帰的に計算することができる。

$$\alpha(i,t) = \sum_j \alpha(j, t-1)a_{ji}b_{ji}(o_t) \tag{2.26}$$

  同様に、$\beta(i,t)$ を最終状態から始まり $o_T, \cdots, o_{t+2}, o_{t+1}$ を生成して状態 i に達する確率（後向き確率）とすると

$$\beta(i,t) = \sum_j \beta(j,t+1) a_{ij} b_{ij}(o_{t+1}) \tag{2.27}$$

となる。

そして F を最終状態の集合、π_i を状態 i の初期確率とすると

$$P(o_1, \cdots, o_T) = \sum_{i \in F} \alpha(i,T) = \sum_i \beta(i,0)\pi_i \tag{2.28}$$

が成立する。更に、次のような関係式が存在する。

$$P(\boldsymbol{o}, x_{t-1} = i, x_t = j) = \alpha(i,t) a_{ij} b_{ij}(o_{t+1}) \beta(j,t+1) \tag{2.29}$$

式 (2.26) から式 (2.29) を Forward-Backward (F-B) アルゴリズムという。

- Viterbi アルゴリズム

HMM による認識アルゴリズムは、DP マッチングの技術を用いて得ることができる。孤立音声の認識には式 (2.22) を用いることができる。連続音声の認識法[39]には 2 段 DP 法や Level Building 法、One Pass DP 法等があるが、2 段 DP 法以外にはここで述べる Viterbi アルゴリズムを使う必要がある。すなわち、式 (2.22) の代わりに

$$P(\boldsymbol{o}) = \max_{\boldsymbol{x}} [\prod_i P(x_i|x_{i-1}) P(o_i|x_{i-1}, x_i)] \tag{2.30}$$

を求めることとなる。これにより、出力系列 $\boldsymbol{x}$ は唯一の状態系列に対応付られ、この状態系列を最適パスと呼ぶ。このアルゴリズムは Viterbi により提案されたことから Viterbi アルゴリズムと呼ばれている (Viterbi アルゴリズムに対し式 (2.22) はトレリスアルゴリズムと呼ばれている)。

なお、この Viterbi アルゴリズムは

$$\log P(\boldsymbol{o}) = \max_{\boldsymbol{x}} [\sum_i \log P(x_i|x_{i-1}) + \log P(o_i|x_{i-1}, x_i)] \tag{2.31}$$

を求めるように変更すれば、乗算が加算の演算で置き換えられるから、計算効率がよくなりアンダーフローも回避できる。

2.4.3 基本 HMM の拡張[39, 52]

(a) 混合連続出力確率分布 HMM

前節では、HMM の出力確率分布が離散分布の場合と連続分布の場合があることを述べた。一般にはよりモデルの精度が良い連続分布が用いられ、分布関数として多次元正規分布を仮定することが多い。しかし、一つの正規分布ではある状態遷移の出力分布を表現できない場合があるので、複数の連続分布の和 (混合分布; continuous mixture densities) で近似することが考えられる。

状態 i からの遷移によってベクトル $\boldsymbol{y}$ が出力される確率 $b_i(\boldsymbol{y})$ は次式で表される。

$$b_i(\boldsymbol{y}) = \sum_{m=1}^{M} c_{im} b_{im}(\boldsymbol{y}) \tag{2.32}$$

但し、c_{im} は m 番目の分布の出現確率を表す分岐確率 (branch probability, mixture weight) で、$b_{im}(\boldsymbol{y})$ は m 番目の確率密度関数を表す。これらには、次の条件が成立する。

$$\sum_{m=1}^{M} c_{im} = 1, \quad \int b_{im}(\boldsymbol{y}) d\boldsymbol{y} = 1 \tag{2.33}$$

通常は $b_{im}(\boldsymbol{y})$ として正規分布 $N(\boldsymbol{\mu}_{im}, \boldsymbol{\sigma}_{im})$ を仮定するが、その場合のパラメータの推定は EM(Expectation-Maximization) アルゴリズムを適用した Baum-Welch(Forward-Backward) アルゴリズムによって求めることができる[39]。しかし、この推定アルゴリズムはかなり計算量を必要であり、再推定の速くて正しい収束のためにはパラメータの初期値が重要だといわれている。そのため、Segmental K-means アルゴリズムという簡易的な手法が用いられることがある[53, 52]。この手法の処理の流れを図 2.3に示す。以下に概要を説明する。

1. パラメータの学習セットと全てのモデルのパラメータの初期値が与えられる。初期パラメータは、ランダムに与えるか適当なモデルに基づいて与える。

2. 学習データの観測ベクトル (出力シンボル) 系列を、現時点のモデル λ に基づいて状態系列に分割 (segmentation) する。この分割は、前述の、最適な状態系列を求める Viterbi アルゴリズムによって行なえる。

3. 各々の状態 j に対応付けられた観測ベクトルを、ユークリッド距離に基づいて M 個のクラスタにクラスタリングする。ここで、M 個のクラスタが M 混合の出力確率分布に対応し、モデルのパラメータはつぎのように更新される。

$$\hat{c}_{jm} = \frac{(\text{状態 } j \text{でクラスタ } m \text{ に対応付けられた観測ベクトル数})}{(\text{状態 } j \text{に対応付けられた観測ベクトル数})}$$

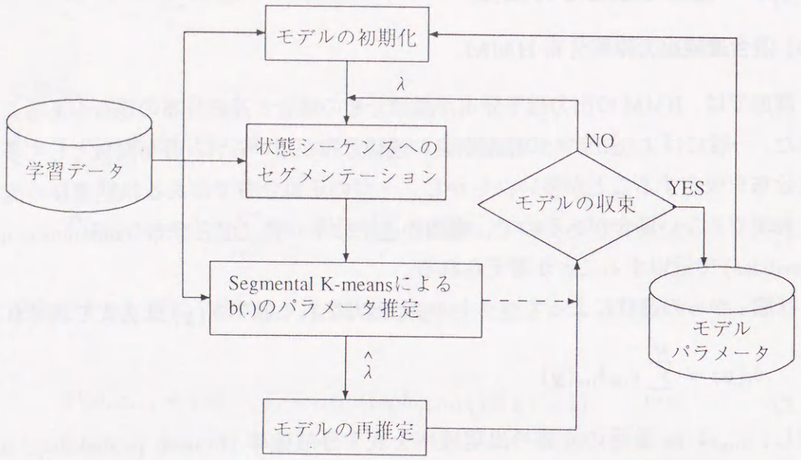


図 2.3: Segmental K-means 学習法[53]

$$\hat{\mu}_{jm} = \begin{pmatrix}\text{状態 } j\text{でクラスタ } m \text{ に対応付けられた観測ベクトル}\\ \text{のサンプル平均}\end{pmatrix} \quad (2.34)$$

$$\hat{\sigma}_{jm} = \begin{pmatrix}\text{状態 } j\text{でクラスタ } m \text{ に対応付けられた観測ベクトル}\\ \text{のサンプル共分散行列}\end{pmatrix}$$

状態遷移確率 a_{ij}は、Viterbi アルゴリズムで得られた最適状態系列に対応して、状態 i から任意の状態へ遷移する回数と状態 i から状態 jに遷移する回数をカウントし、その比によって求めることができる。

4. 新しく得られたパラメータによって更新されたモデル$\hat{\lambda}$を用いて、正式な再推定法によって全てのモデルのパラメータを再推定する。その結果得られたモデルを以前のモデルと比較し、収束の条件が満たされるまで以上の手続きを繰り返す。モデルの比較は、HMM の統計的な類似度を反映したモデル間の距離スコアや訓練サンプルに対するモデルの尤度の差を用いる。

(b) HMM の連結法

HMM は一般にサブワード (subword) 単位の音響的モデルとして用いられるため、任意の単語や、より大きな単位のモデルは、その音声表記の辞書に従って連結することによって得られる。図2.4は、単語「メール」の発音に対応する音節の並び/me/,/e/,/ru/から、単語レベルのモデルが構成される例を示している。この例のように、モデルの終了状態と初期状態の間を、シンボルの出力がない null アークで接続する。単語や文

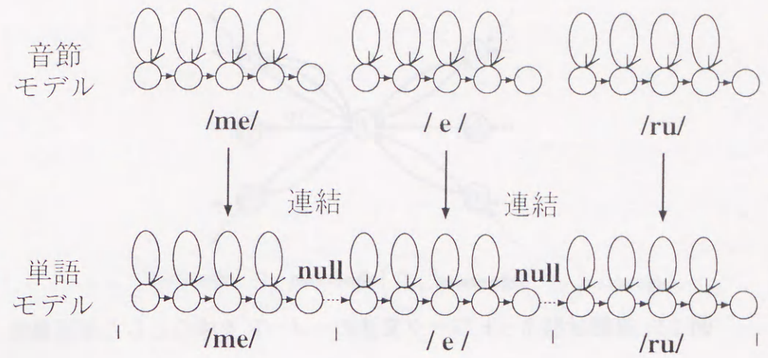


図 2.4: HMM の連結

の単位の音声による HMM の学習では、このような HMM の連結を行なうことによってあらかじめ明示的にサブワード単位にセグメンテーションを行なう手間が省けるため、大量のラベルが無い学習用音声データからパラメータの再推定を行なう場合には、そのようにして学習（いわゆる連結学習）を行なうことが多い[54, 75][1]。

連続音声認識では、言語モデルの制約によってある単語に接続が許される単語が幾つもあり得るので、上述のような連結を全て一つの有限状態ネットワークに変換しておくことは困難である場合が多い。そこで、実際には null アークによる接続と同等な処理を動的に実現することがある。例として、有限状態ネットワーク文法の制約を用いた連続単語認識の場合の、図 2.5に示すようなあるネットワークのノード p を中心とした単語の HMM の尤度計算について考える[2]。ノード s_1, s_2, s_3から p への遷移に対応する各単語のアークは遷移先において全てマージしているので、ノード p での処理はそれらの単語のアークに対応するパスの内で最大累積尤度のパスを選ぶことである。従って、ノード p に入る複数のアーク上の単語の HMM から後続するアーク上のある単語 wの HMM への接続に関してのフレーム i における計算は、ノード p までの最適パスの最大累積尤度 $P_p(i)$ を求め、単語 wの HMM の初期状態の累積尤度の更新において $P_p(i)$ を考慮することで実現される[3]。

(c) 継続時間制御の組み込み

図 2.2に示したような基本的な HMM の構造では、ある状態 i に n 時刻留まる確率は、

[1]単語間の無音区間などを考慮する場合は、null アークと並行して“無音モデル” を連結するとよい。
[2]前述のように HMM による連続単語認識では Viterbi アルゴリズムを基本とする必要がある。
[3]詳細は 2.4.4節の連続音声認識アルゴリズムを参照。

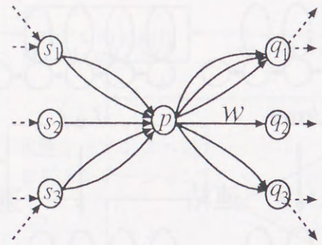


図 2.5: 有限状態ネットワーク文法の一ノードを中心とした単語遷移

$$d(n) = a_{ii}^{n-1}(1 - a_{ii}) \tag{2.35}$$

で与えられる。したがって、指数関数的に確率は減少するため、音声の定常区間の継続時間を表すモデルとして妥当であるとはいえない。そのような欠点を解決するためには、状態数を多くする方法や、後処理による方法、継続時間分布モデルを導入する方法などがある[39]。本研究で用いる音声認識システムでは継続時間分布モデルを用いるので、この方法について簡単に述べる。

継続時間分布を持つ HMM の構造を図 2.6に示す。継続時間分布 $d_i(\tau)$ は、離散分布モデル、またはポアソン分布やガンマ分布のような連続分布モデルが一般に用いられる。継続時間分布を用いる場合も、HMM のパラメータ推定には Baum-Welch の推定アルゴリズムをそのまま適用できる[39]。認識に Viterbi アルゴリズムを使用する場合は、簡単にアルゴリズムの中に継続時間分布の制限を導入することができる。この場合、状態 i、時刻 t における対数累積確率は次式で表される。

$$\begin{aligned}\log f(i,t) \;=\; & \max_{j,j'}[\max_{\tau}\{\log f(j,t-\tau) + \log a_{ji} + \alpha \log d_i(\tau) \\ & + \sum_{k=1}^{\tau} \log b_{ji}(y_{t+1-k})\}, \log f(j',t) + \log a_{j',i}]\end{aligned} \tag{2.36}$$

但し、右辺の第 2 項は null 遷移に対応するものである。このように継続時間制御を加えると、通常の Viterbi アルゴリズムに対して計算量は増加する。

2.4.4 HMM による連続音声認識アルゴリズム - One-Pass Viterbi 法 -[39, 52]

音声認識システムにおいて言語モデルが使用される場合、言語処理レベルにおける目的は、言語的な制約のもとで最適な単語列を探索することである。そのためには、前述のような幾通りかのシステムの構成モデルが考えられる。音声認識レベルでより高

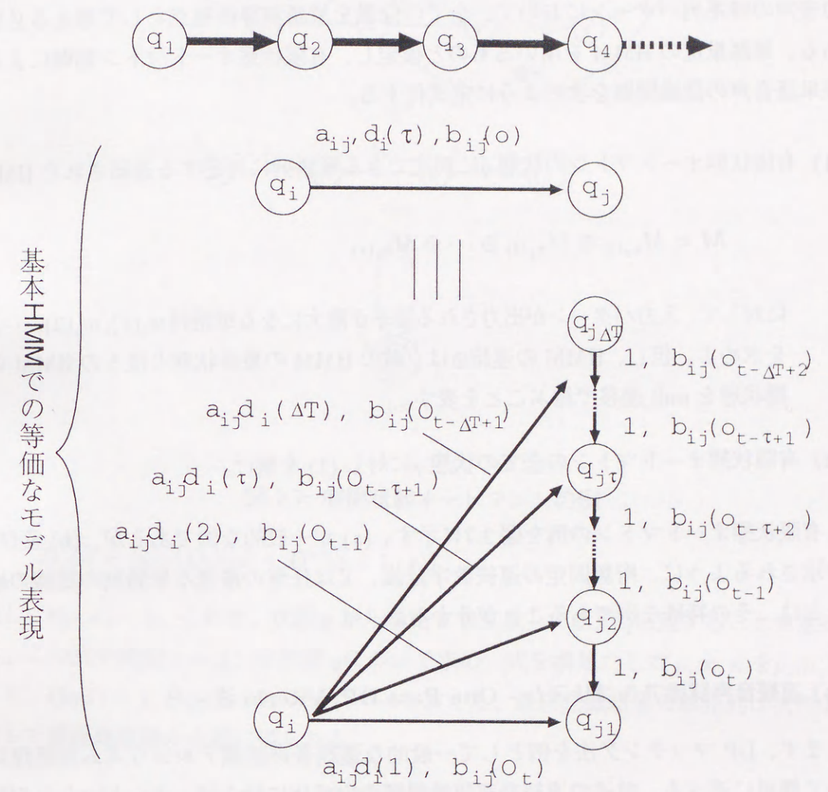


図 2.6: 継続時間制御付き HMM

度な知識を導入することは、探索問題としての最適解を得ることを保証するためにも望ましい。ここでは、有限状態オートマトンという小規模のタスクを記述するのに適した言語モデルを使って、構文制御による連続音声認識を実現するアルゴリズムを示す。

(a) 連続音声認識の定式化

連続音声認識においては、単語境界（音節や音韻と考えてもよい）に関しては、入力音声の時系列パターンにおいて、全ての位置を単語境界候補点として考える必要がある。単語単位の HMM を用いるものと仮定し、有限状態オートマトン制御による連続単語音声の認識問題を次のように定式化する。

(1) 有限状態オートマトンの状態 qに到達できる単語列に対応する連結された HMM

$$M = M_{w_q(1)} \oplus M_{w_q(2)} \oplus \cdots \oplus M_{w_q(x)}$$

に対して、入力パターンが出力される確率が最大になる単語列 $w_q(1), w_q(2), \cdots, w_q(x)$ を求めよ。但し、HMM の連結$\oplus$は、前の HMM の最終状態と後ろの HMM の初期状態を null 遷移で結ぶことを表す。

(2) 有限状態オートマトンの全ての状態 qに対し (1) を解け。

有限状態オートマトンの例を図 2.7に示す。(a) が一般的な例であるが、(b) 及び (c) で示されるように、桁数固定の連続数字認識、又は任意の最適な単語列の認識の場合などは、その特殊な例であることが分かる。

(b) 連続音声認識アルゴリズム - One Pass DP/Viterbi 法 -

まず、DP マッチング法を例として一般的な連続音声認識アルゴリズムの原理について簡単に述べる。前述の連続音声認識問題の定式化に従えば、オートマトンの制約に基づく連続音声認識は、入力パターン $\boldsymbol{T} = \boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_I$（$\boldsymbol{a}_i$は音声の i フレーム目の観測ベクトル）と、オートマトンの最終状態 Q に達する任意の可能な単語列に対応する単語標準パターン系列 $\boldsymbol{R}_Q$との間で、累積マッチング距離が最小となる単語列を見つける問題として考えることができる。ここで、単語 n の標準パターン $\boldsymbol{R}^n$と入力パターンの部分パターン $\boldsymbol{a}_{m+1}, \boldsymbol{a}_{m+2}, \cdots, \boldsymbol{a}_i$との間の DP マッチングの累積距離を $\boldsymbol{D}^n(m+1:i)$ と表すことにする。先ほどの問題を一般的にして、入力パターンが $1 \sim i$ フレームで注目するオートマトンの状態を qと考えると、最小となる累積マッチング距離は次のような漸化式で与えられる。

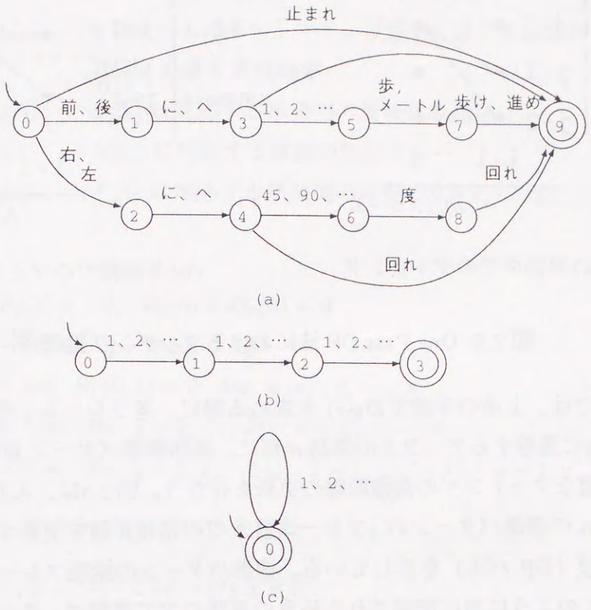


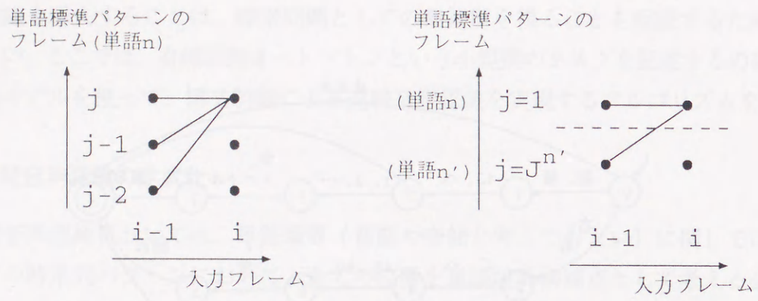
図 2.7: 有限状態オートマトンの例

$$\boldsymbol{D}_q(i) = \min_{p,m,n}\{\boldsymbol{D}_p(m) + \boldsymbol{D}^n(m+1:i)\} \tag{2.37}$$

但し、$\delta(p,n) = q$。これは、状態 p から単語 n を生成して状態 qに達することを意味する。この式を時刻 $1 \sim I$、全状態 qについて求め、式を満足たした p, m, n を$\hat{p}, \hat{m}, \hat{n}$として、$\boldsymbol{Q}q(i) = \hat{p}, \boldsymbol{B}q(i) = \hat{m}, \boldsymbol{N}q(i) = \hat{n}$とおくと、最終認識結果の単語列は次のようにして最後尾単語から順に得られる。

1. $i \longleftarrow I. \quad q \longleftarrow arg\min_{q \in F} \boldsymbol{D}_q(i).$
2. $\boldsymbol{N}_q(i)$ を認識結果として出力。
   $q \longleftarrow \boldsymbol{Q}_q(i). \quad i \longleftarrow \boldsymbol{B}_q(i).$
   $i = 0$ なら終了。　$i \neq 0$ なら 2 番へ。

上述の手順を効率的に行ない最適解を求める方法として、One Pass DP 法が提案されている[20, 21]。One Pass DP 法はフレーム同期的アルゴリズムで、DP マッチング法においては最も一般的な手法の一つである。HMM を用いた認識においてこのアルゴリズムの原理を適用する場合は、Viterbi アルゴリズムによって同様に実現できる。One



(a)単語中でのマッチング　　(b)単語間でのマッチング

図 2.8: One Pass DP 法におけるマッチング処理[39]

Pass DP 法では、上述の手順で$\boldsymbol{D}q(i)$ を求める際に、各フレーム i で各々のオートマトンの状態 qに遷移するアーク上の単語 n 毎に、単語標準パターン $\boldsymbol{R}^n$と入力パターンとの間の最適なマッチングの累積距離の更新を行なう。図2.8は、入力側 i フレーム処理時に単語 n の標準パターンの jフレーム目までの累積距離を更新するときの最適なパスの選択肢（DP パス）を示している。標準パターンの始端フレーム $j=1$ のときだけ、図 (b) のように前に接続される任意の単語の中で累積マッチング距離が最小の単語 n'とのパスを考慮し、単語内でのマッチング (図 (a)) とは異なった処理を行なう。HMM を用いる場合には、図2.8をViterbi アルゴリズムに置き換えて考えることによって同様なアルゴリズムとなる。HMM に基づく連続音声認識アルゴリズムを以下に示す[39, 20, 21]。

(c) HMM 法に基づくオートマトン制御 One Pass Viterbi 法

＜記号の定義＞

N: 単語数（語彙数）

n: 単語名

I: 入力音声のフレーム長

J^n: 単語 n の HMM の状態数

j: HMM の状態番号（$1 \sim J^n$）

q: オートマトンの状態番号（0∼Q）

$P_q^n(i,j)$: 単語 n，オートマトン状態 q，HMM 状態 j，i フレームまでの最適パスの累積確率

$B_q^n(i,j)$: 単語 n，オートマトン状態 q，HMM 状態 j，i フレームまでの最適パスのバックポインタ

$P_q(i)$: i フレームでオートマトン状態 qに達する単語列の HMM の最大累積確率

$N_q(i)$: $P_q(i)$ に対応する単語列の最後尾単語名

$B_q(i)$: $N_q(i)$ に対応する単語の開始フレーム − 1

$Q_q(i)$: $P_q(i)$ に対応する単語列の状態 qの直前の状態

(1) 初期条件 $P_0(0)=1.0, \quad B_0(0)=Q_0(0)=0$

$P_q(0)=-\infty \quad \text{for} \quad q=1,2,\cdots,Q$

$P_0^n(0,1)=1.0,\ B_0^n(0,1)=0 \quad \text{for} \quad n=1,2,\cdots,N$

$P_0^n(0,j)=-\infty \quad \text{for} \quad j=2,\cdots,J^n;\ \ n=1,2,\cdots,N$

$P_q^n(0,j)=-\infty \quad \text{for} \quad q=1,2,\cdots,Q;\ \ j=1,2,\cdots,J^n;\ \ n=1,2,\cdots,N$

(2) $i=1,2,\cdots,I$ について (3)∼(7) を実行

(3) $q=1,2,\cdots,Q$ について (4)∼(7) を実行

(4) $n=1,2,\cdots,N$ について (5)∼(6) を実行

(5) $P_q^n(i-1,0)=\max_p P_p(i-1)$

$B_q^n(i-1,0)=i-1 \qquad$ 但し、$\delta(p,n)=q$

$\hat{j}=\text{argmax}_j\{P_q^n(i-1,0),\ \ P_q^n(i-1,1)\}$

$P_q^n(i-1,1)=P_q^n(i-1,\hat{j})$

$B_q^n(i-1,1)=B_q^n(i-1,\hat{j})$

(6) $j=1,2,\cdots,J^n$ について

$\hat{h}=\text{argmax}_{h(null遷移を除く)}P_q^n(i-1,h)a_{hj}\cdot b_{hj}(o_i)$

$\hat{k}=\text{argmax}_{k(null遷移)}P_q^n(i,k)a_{kj}$

$P_q^n(i,j)=\max\{P_q^n(i-1,\hat{h})a_{\hat{h}j}\cdot b_{\hat{h}j}(o_i),\ \ P_q^n(i,\hat{k})a_{\hat{k}j}\}$

$B_q^n(i,j) = B_q^n(i-1,\hat{h})$; $P_q^n(i,j) = P_q^n(i-1,\hat{h})a_{\hat{h}j} \cdot b_{\hat{h}j}(o_i)$ のとき

$B_q^n(i,j) = B_q^n(i,\hat{k})$; $P_q^n(i,j) = P_q^n(i,\hat{k})a_{\hat{k}j}$ のとき

(7) $\hat{n} = \mathrm{argmax}_n P_q^n(i, J^n)$

$\hat{p} = \mathrm{argmax}_p P_p(B_q^{\hat{n}}(i, J^{\hat{n}}))$ 但し、$\delta(p,\hat{n}) = q$

$P_q(i) = P_q^{\hat{n}}(i, J^{\hat{n}})$

$B_q(i) = B_q^{\hat{n}}(i, J^{\hat{n}})$

$N_q(i) = \hat{n}$

$Q_q(i) = \hat{p}$

(8) バックトレースによる単語列判定処理

2.5 タスクの言語的な複雑性の尺度[39, 42]

音声認識システムの評価を行なう場合、そのシステムが前提としている様々な制約を考慮しなければならない。特に、システムが音声入力において仮定する文を文法などによって規定する場合、その言語的な制約の大きさを知ることが必要となる。言語的な制約が緩いほど音声認識は困難になり、そのような制約が緩い音声入力を前提としたタスクはより複雑であるといえる。ここでは、言語モデルに対する複雑性を知る尺度としてよく用いられているものを簡単に説明する[39]。

2.5.1 静的分岐数と平均ファンアウト数

簡単のために、言語Lが有限状態オートマトン（正規文法）で表現されているとする。このとき$\pi(j)$を状態jの出現確率、$n(j)$を状態jから遷移できる単語数とすれば、静的分岐数（static branching factor; F_S）と平均ファンアウト数（fanout; F_A）は次式によって定義される。

$$F_S(L) = \frac{\sum_j n(j)}{\sum_j 1} \tag{2.38}$$

$$F_A(L) = \sum_j \pi(j) n(j) \tag{2.39}$$

平均ファンアウト数は各状態を出現確率が等確率のとき、静的分岐数に等しくなる。F_S、F_Aはタスクの表現法によって異なる値を持つ[39]。

2.5.2 エントロピー

言語Lにおいて、単語列（音節列，音韻列）$\boldsymbol{w}_1^k = w_1, w_2, \cdots, w_k$の出現確率を$P(\boldsymbol{w}_1^k)$とすれば、言語$L$のエントロピーは次式で定義される。

$$H_0(L) = -\sum_{\boldsymbol{w}_1^k} P(\boldsymbol{w}_1^k) \log_2 P(\boldsymbol{w}_1^k) \tag{2.40}$$

単語列の代わりに音韻列を用いても$H_0(L)$の値は不変である。また、1単位当たりのエントロピーは、

$$H(L) = -\sum_{\boldsymbol{w}_1^k} \frac{1}{K} P(\boldsymbol{w}_1^k) \log_2 P(\boldsymbol{w}_1^k) \tag{2.41}$$

である。

言語Lが有限状態オートマトンで表現される場合には、エントロピーは次のように定義できる。$P(w|j)$を状態jで単語（音節、音韻）wを出力する確率とすれば、状態jでの1単位当たりのエントロピーは、

$$H(w|j) = -\sum_{w_1^k} P(w|j) \log_2 P(w|j) \tag{2.42}$$

各状態についての期待値の和を取れば、

$$H(L) = \sum_j \pi(j) H(w|j) \tag{2.43}$$

となる。

言語Lが文脈自由文法等で表現される場合でも文の長さ分布を実際のサンプルから算出し、等しい文長の文は互いに等確率で生起すると仮定すれば、次のようにしてエントロピーを求めることができる[24]。

P_k：文が長さkである確率

N_k：言語Lによって生成される長さkの文の総数

$$\begin{aligned} H_0(L) &= -\sum_{\boldsymbol{w}_1^k} P(\boldsymbol{w}_1^k) \log_2 P(\boldsymbol{w}_1^k) = -\sum_{\boldsymbol{w}_1^k} \frac{P_k}{N_k} \log_2 \frac{P_k}{N_k} \\ &= -\sum_k N_k \frac{P_k}{N_k} \log_2 \frac{P_k}{N_k} = -\sum_k P_k \log_2 \frac{P_k}{N_k} \end{aligned} \tag{2.44}$$

1単位当たりのエントロピーは

$$H(L) = -\sum_k N_k \frac{1}{k}\frac{P_k}{N_k}\log_2\frac{P_k}{N_k} = -\sum_k \frac{P_k}{k}\log_2\frac{P_k}{N_k} \tag{2.45}$$

以上、文法を用いた言語モデルに対しての代表的なものを示したが、そのほかに言語モデルとして単語・品詞の2,3字組確率を用いて単語列（文）の共起確率を近似するbigram、trigramがある。

単語の共起確率のエントロピーを以下の式で定義する。

$$H(L) = -\sum_{ij} P(w_i w_j)\log_2 P(w_j|w_i) \quad : \quad \text{bigram} \tag{2.46}$$

$$H(L) = -\sum_{ijk} P(w_i w_j w_k)\log_2 P(w_k|w_i w_j) \quad : \quad \text{trigram} \tag{2.47}$$

ここで、$P(w_i w_j)$、$P(w_i w_j w_k)$ は単語の共起確率、$P(w_j|w_i)$、$P(w_k|w_i w_j)$ は条件付確率である。

2.5.3 パープレキシティ

符号化定理によれば、言語 L の1単位当たりのエントロピーが $H(L)$ なら、次の単位語を決定するのに平均 $H(L)$ 回のyes/no質問を繰り返さなければならない。言い換えれば、$2^{H(L)}$ 個の等出現確率の単位語から1単位語を決定することになる。これは、情報理論的な意味での平均分岐数であり、パープレキシティと呼んでいる。これを $F_p(L)$ と記せば次式で与えられる。

$$F_p(L) = 2^{H(L)}$$

パープレキシティは、文法の記述法に依存しない優れた特徴を持っている[39]。

2.5.4 テストセットパープレキシティ

一般にパープレキシティ $F(L)$ は、テスト文が少ないときはテスト文によって認識の難易差が存在するので、テスト文集合に対するパープレキシティを求める必要がある。これをテストセットパープレキシティと呼ぶ。文認識の際に、単語の文頭になる確率と文末になる確率を考慮する必要性をなくすために、一文が"ピリオド"から始まり"ピリオド"で終ると定義する[25]。テストセットパープレキシティは、入力文を "$\cdot w_1 w_2 \cdots w_{n-1} w_n \cdot$" とすると、以下の式で定義される。

一般式

$$F_T(p) = \left(\frac{1}{P(w_1|\cdot)} \times \frac{1}{P(w_2|w_1)} \times \frac{1}{P(w_3|w_1 w_2)} \times \cdots \times \frac{1}{P(w_n|w_1 w_2 \cdots w_{n-1})} \times \frac{1}{P(\cdot|w_1 w_2 \cdots w_n)}\right)^{\frac{1}{n+1}} \tag{2.48}$$

bigramのときは、

$$F_T(p) = \left(\frac{1}{P(w_1|\cdot)} \times \frac{1}{P(w_2|w_1)} \times \cdots \times \frac{1}{P(w_n|w_{n-1})} \times \frac{1}{P(\cdot|w_n)}\right)^{\frac{1}{n+1}} \tag{2.49}$$

trigramのときは、

$$F_T(p) = \left(\frac{1}{P(w_1|\cdot\cdot)} \times \frac{1}{P(w_2|\cdot w_1)} \times \cdots \times \frac{1}{P(\cdot|w_{n-1} w_n)} \times \frac{1}{P(\cdot|w_n \cdot)}\right)^{\frac{1}{n+1}} \tag{2.50}$$

単語の出現確率を考慮しないで予測される単語数（分岐数）のみ考慮する場合（単語対を使用する場合）は、$w_1 w_2 \cdots w_{i-1}$ の次に予測される単語数を c_i とすれば、

$$F(p) = (c_1 \cdot c_2 \cdots c_n)^{\frac{1}{n}} \tag{2.51}$$

となる。つまり、分岐数の幾何（相乗）平均となる。

ここで、各単語A,B,Cが独立にそれぞれ確率2/3,1/6,1/6で生起する記憶のない情報源のエントロピー、パープレキシティ、テストセットパープレキシティを求めてみる。

$$H(p) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{6}\log_2\frac{1}{6} - \frac{1}{6}\log_2\frac{1}{6} = 1.25 \tag{2.52}$$

$$F(p) = 2^{H(p)} = 2.38 \tag{2.53}$$

$6n$ 単語からなるテスト文集合では、Aが $4n$ 回出現し、Bが n 回、Cが n 回出現するとした場合（n が大きくなると大数の法則によりこのことが成立する）のテストセットパープレキシティは、

$$F_T(p) = \left\{\left(\frac{3}{2}\right)^{4n} \times 6^n \times 6^n\right\}^{\frac{1}{6n}} = 2.38 \tag{2.54}$$

となる。$F(p)$ と $F_T(p)$ は、テスト文の中の単語の出現回数が各単語の生起確率に比例しているならば（テスト文集合が多くなれば比例する）等しくなるはずであるので、この結果よりここで定義したテストセットパープレキシティの式は、妥当であることが理解できよう。

第 3 章

文脈自由文法制御による連続音声認識システム - SPOJUS-SYNO -

3.1 はじめに

文認識を対象とした連続音声認識においては言語情報を用いるのが一般的であるが、文脈自由文法を言語モデルとして用いる場合は、連続音声認識の処理にその言語的制約を組み込むのは容易ではない。そのため、一般に単語や音韻のスポッティングによって得られる候補を中間的に出力し、構文解析法によって構文的に正しい文候補を求める方法が採られる。著者の研究室で以前に開発された連続音声認識システム SPOJUS-SYNO(SPOken Japanese Understanding System - SYNtax Oriented)[22]は、ワードスポッティング法によって得られるワードラティスから構文解析法によって文候補を求めているため、そのような階層的な構成方式に属する。このシステムでは、HMM による音響モデル及び言語モデルのそれぞれについて、改良や評価が行われ（SPOJUS-SYNO·II）[41, 42]、「UNIX-QA」に関するタスクにおいて、男性話者 6 名に対する平均文認識率 80.7 %が得られている。

しかし、従来の階層型のシステムは、音声認識部及び言語処理部の評価が独立して行えるという利点がある一方で、中間的な出力を介するため一般に精度が劣化する問題がある。そこで、本研究では従来のシステムをより高精度化することを目的として、統語処理と音声認識処理を統合すると同時に、認識システムの高速化を図っている（SPOJUS-SYNO·X）。なお、文脈自由文法による統語処理と音声認識処理の統合化手法として、CYK 法に基づく方法[44]、LR パーザに基づく方法[28, 29]、チャートパーザに基づく方法[45]などがある。我々の方法は Earley のパーザに基づく方法[26]である。本章は、はじめに従来のシステムの概略について触れたあと、本研究で開発した 2 種類の統合化

されたシステムの認識アルゴリズムと、高速化のための枝刈り手法や構文解析の効率化について述べる。最後に、これまでに開発されたシステムの評価実験の結果を示す。

3.2 従来の日本語連続音声認識システム - SPOJUS-SYNO I/II -

連続音声認識・理解システム SPOJUS-SYNO I/II[22]は、著者の研究室において中規模のタスクを対象としたシステムとして開発されてきたものである。本システムは、入力音声からワードスポッティング法によってセグメンテーションや単語の曖昧さを含む複数の単語仮説を出力し（単語ラティス）、その結果を構文情報を用いて解析して文法的に正しい文を認識する。ワードスポッティング部の音声との照合の処理は単語単位で行なわれるが、単語の音響的モデルは音節単位の HMM の連結によって表現するため、実際には単語辞書から自動的に構成される音節 HMM の系列に基づいて照合を行なっている。音響モデルの単位として音節を用いているため、語彙の追加や変更に柔軟に対処可能であり、単語辞書と文法の変更のみで他のタスクにも適用できるようになっている。

図 3.1 にシステムのブロック図を示す。システムが持つ言語情報は、構文・意味的な情報を記述した文脈自由文法と、単語の音節表記のリストを持った単語辞書からなっている。機能的には、このシステムは音響処理部、ワードスポッティング部、構文解析部の 3 つに大別される。ワードスポッティング部では、単語辞書に従って音節単位の HMM を連結し、単語単位の HMM を自動構成する。例えば「私（watasi)」の HMM は、/wa/,/ta/,/si/の 3 音節の HMM の連結で構成される。この単語単位で構成された HMM を用いてワードスポッティングを行ない、入力音声を単語ラティスに変換する。単語ラティスとは、入力音声中で存在しそうな単語候補をすべて出力した表現形式で、通常（始端位置、終端位置、単語名、スコア）の 4 つ組で表現される。構文解析部は構文知識と単語ラティスのスコアに基づいて解析可能な単語列を単語ラティス中から選択し、入力文を再現する部分である（ラティスパージング)。このシステムでは、left-to-right & top-down 型構文解析法を採用し、単語列を構築する際に部分文が爆発的に増加するのを防ぐためにビームサーチ法を導入している。このシステムの詳細については文献[22]を参照されたい。

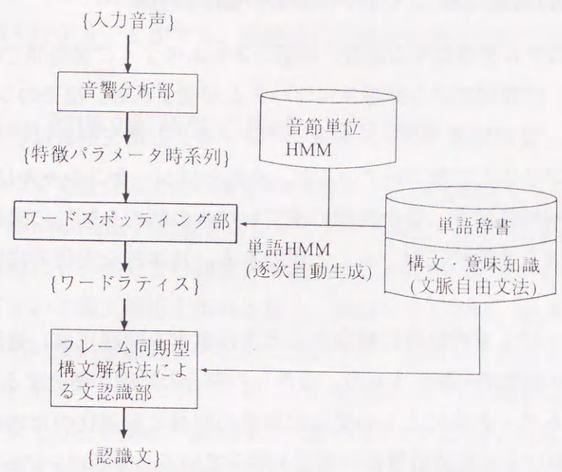


図 3.1: SPOJUS-SYNO・II の概略図

3.3 音声処理と言語処理の統合

連続音声認識／理解システムでは、言語情報の利用が重要であることは明らかであるが、従来の多くのシステムでは音声処理と言語処理の間が密に結合されておらず、一般に言語処理部は、簡単な言語モデルを用いた音声認識の出力を元にして処理するようになっている。しかし、最近は音声言語処理としての音声認識技術が重要視され、言語情報を効果的に利用するために処理を統合化する技術が注目されている[6]。一方で、処理をあえて統合化しないものとして、N-best パラダイムがある[18, 16]。一般に、N-best 方式の音声認識では、統計的な言語モデル (例えば、bigram, trigram など) を用いて[1]可能性の高い候補を多く出力し、その結果得られる候補のリストに対してより高次の言語情報を用いた解析を行ない再評価する。

より高次の言語情報に基づく言語的制約を音声処理に統合するには、一般に構文・意味や文脈レベルの知識を効率的に扱える認識アルゴリズムを考える必要がある。ここでは、SPOJUS-SYNO の改良における構文や文脈レベルでの言語的制約の統合化の方法について述べる。

[1]低次の言語処理が統合化されているとも解釈できる。

3.3.1 言語知識に基づく音声処理の動的制御

音声認識システムを構築する場合、対話システムのように実時間での応答が必要な場合を考えると、時間同期的な認識アルゴリズムが望まれる。従来のシステム SPOJUS-SYNO-I/II は、音声処理（ワードスポッティング）と構文解析は、どちらもフレーム同期的なアルゴリズムに基づいているが、それらはシーケンシャルに用いられていた。そこで、構文解析処理を、音声処理の構文レベルの制約を動的に求める予測的なパーザとして、両者を並列的に用いることができる。具体的な実現方法については次節で述べる。

予測的なパーザを音声処理に統合するときの重大な問題点は、処理が進むに従って部分文の仮説が指数的に膨れ上がり、音声との照合の処理が増大することである。本研究では、ビームサーチ法による必要な記憶量の削減と高速化のための検討に加え、音声との照合における近似的計算法の実現を試みている。後者は、ワードスポッティング法に基づいており、音声との照合のための計算量を語彙数のオーダーに抑えることができる方法である。この方法によって統合化されたシステム (SPOJUS-SYNO-III) については、3.4節で述べる。一方、2.4.4節で述べた One Pass Viterbi 法により統合し、ビームサーチ法によって準最適な照合を行なうシステム (SPOJUS-SYNO-X) については、3.5節で述べる。

音声対話システムでは、言語情報として文脈やプラグマティクスなどの利用も有効と考えられる。従来のシステム SPOJUS-SYNO は、音声対話システムの音声認識フロントエンドとして用いらているが、これまでは文脈レベルの言語情報の利用はしていなかった。しかし、SPOJUS-SYNO と接続する対話システムが最近改良され[51]、ユーザの次発話を予測する機能が追加された。そこで予測の情報を音声認識部で利用するために行なった改良について述べる。

ユーザの次発話予測では、対話の進行に従ってユーザが用いる単語を予測し、直前のシステムの発話の内容と文型に基づいてユーザ発話の構文的な予測を行なう。単語レベルの予測と構文的な予測の情報は、それぞれ、予測される単語カテゴリ名や構文カテゴリ名とその予測確率で与えられる。単語カテゴリ名は、あらかじめ音声認識部が持つ語彙リストの一部の単語集合に対応付けておく。音声認識部は、次発話予測が行なわれるたびに、予測された単語カテゴリに属する単語だけからなる単語リストを再構成し、認識のために用いられる語彙を制限する。予測される構文カテゴリ名については、文脈自由文法の一部の書き換え規則の集合に対応付けておく[2]。構文予測においても同様に、音声認識部は、次発話予測が行なわれるたびに予測された構文カテゴリに属する書き換え規則の適用確率を変更し、認識のために用いられる書き換え規則を制限する。このような次発話予測の情報の利用により、予測されたユーザの発話内容を仮定した認識を行なうことができ、誤認識の改善や処理の効率化が期待できる[51]。

[2]付録 B.1を参照。書き換え規則の右辺の最後に '/' に続けて書かれている単語が構文カテゴリ名。

3.3.2 Earley 型構文解析法に基づく構文制御

Earley 法に基づく文脈自由文法の構文解析法は、breadth-first & top-down 型の解析法で、文頭から left-to-right に処理を行なうことができる。従って、Earley 法を音声処理と統合する場合にも、部分文の仮説を予測するために用いることは比較的容易である。音声認識において構文解析を用いる場合、問題となるのは、音声との照合の結果得られる複数のあいまいな部分文の仮説についても、並行して処理する必要があるということである。そこで、それぞれの部分文の仮説についての解析の状態を覚え、効率的に解析を行なう方法を採る。基本的に、従来のシステム SPOJUS-SYNO-I/II において用いられる構文解析法[26]に基づいているが、以下に簡単に述べる。

文法は、図3.2のように一般的な文脈自由文法の記法により表現される。但し、@で始まるストリングは非終端記号を表し、＊で始まるストリングは非終端記号の一種であるが文法的に同一な終端記号の集合を表している。後者をワードクラスと呼び[3]、この書換え規則は一般の書換え規則と区別し、終端記号のように扱う。

構文解析法は、Earley のトップダウン型アルゴリズムに基づくものであるが、音声照合を行なう複数の部分文仮説について、部分文の後続単語の予測をフレーム同期処理の中で効率よく行うために、途中解析結果の記憶方法を工夫する。文法の書換え規則には、記号の位置に対して固有の番号を割り当て（例えば、図3.2で番号 5(=5+0) は @S を、番号 12(=10+2) は@NP2 を示す）、構文解析の過程（書換え規則の適用履歴）を数字のストリングで表現する。

例えば、図3.2の文法を用いた場合に、"*NOUN" というワードクラスだけの部分文（I, JOHN, MAN などの一単語の部分文）の導出過程を考えると、

@S → @NP @VP
@S → @NP2 @VP
@S → *NOUN @VP

となる。そこで、この文法の適用の過程を文法上の位置による表現（数字のストリング）で表すと、次のような文法の適用履歴によって表現される。

"5" → "6" → "6 15" → "6 16" → "6 16 25" → "6 16 26" → *NOUN の予測

こうして表現される数字のストリング（以後、"文法上のパス" と呼ぶ）は、単語（部分文）を予測した段階での構文解析の途中結果を記憶するために用いられる。上の例

[3]一般には前終端記号という。

```
        0          1       2       3 ···
 5  @S     →  @NP     @VP
10  @NP    →  *DET    @NP2
15  @NP    →  @NP2
20  @NP2   →  *ADJ    @NP2
25  @NP2   →  *NOUN
30  @VP    →  *VERB
35  @VP    →  *VERB   @NP

*NOUN  → I
*NOUN  → JOHN
*NOUN  → MAN
*NOUN  → TENNIS
*VERB  → KNOW
*VERB  → PLAY
*DET   → A
*DET   → THE
*ADJ   → BIG
*ADJ   → YOUNG
```

図 3.2: 文脈自由文法の例

では、“*NOUN” を予測した段階での部分文の解析の状態は、“6 16 26” という文法上のパスで表される。ここで、更に、上述の部分文（“*NOUN” という単語）の後続単語の予測を考えると、既に得られている解析の状態（文法上のパス）をもとに、次のような一連の文法の適用によって求めることができる。

$$\text{“6 16 26”} \rightarrow \text{“6 16”} \rightarrow \text{“6”} \rightarrow \text{“7”} \begin{cases} \rightarrow \text{“7 30”} \rightarrow \text{“7 31”} \rightarrow \text{*VERB の予測} \\ \text{または} \\ \rightarrow \text{“7 35”} \rightarrow \text{“7 36”} \rightarrow \text{*VERB の予測} \end{cases}$$

ある部分文に対応する文法上のパスが与えられた時、上述の例のようにその後に隣接可能な単語を予測するためのアルゴリズムを以下に示す。

単語予測アルゴリズム

Step 1. パスリストに、与えられた文法上のパスを入れる。

Step 2. パスリストが空きであれば終了。そうでなければ、パスリスト中から文法上のパスを一つ選択し、その文法上のパスの右端の数値を 1 増加する（右隣りの文法上の位置を示す）。この数値が示す文法上の記号に基づいて、以下の処理を行なう。

- もし、この記号が終端記号なら、その終端記号を予測し、その文法上のパスを返す。Step 2 を繰り返す。
- もし、この記号がワードクラスを示すなら、そのワードクラスに属する単語を予測し、文法上のパスを返す。Step 2 を繰り返す。
- もし、この記号が非終端記号を示すなら、この非終端記号を左辺にもつ書き換え規則を探し、その位置 (head) を文法上のパスの右端に連結する。これは、同時に複数起こり得るが、得られた全ての文法上のパスをパスリストに入れる。Step 2 を繰り返す。
- もし空なら、文法上のパスの右端の数値を取り除き、パスリストに入れる。Step 2 を繰り返す。

このアルゴリズムでは、左再帰規則がある場合に文法上のパスが無限長になってしまう問題があるが、パスの長さに制限を設けることによって対処する。

実際に、音声処理と上述の解析法を組み合わせて処理を進めるときには、文頭から順に部分文の仮説を予測し、音声との照合を行なう。このとき、解析の状態、すなわち文法上のパスが同じ部分文の仮説については、Viterbi アルゴリズムに基づく音声照合アルゴリズムを用いる場合、その後の照合が同一なのでまとめることができ、計算を効率化できる。そこで、異なる文法上のパスを有限状態オートマトン（ネットワーク）の一状態にそれぞれ対応させ、状態間の弧に単語を対応させたような形式で予測された範囲の構文情報を記憶するとよい。これらの実現法については 3.5.1節で述べる。

3.4 ワードスポッティング法に基づいた統合化 - SPOJUS-SYNO III -

ここでは、拡張連続 DP 法[27]の基本的な考え方に基づいて、ワードスポッティング手法を基盤として連続音声認識アルゴリズムを構成する方法について述べる[48]。連続発声された音声から単語系列を求めるアルゴリズムとしては、DP マッチング法に基づいた効率的な方法がいくつか提案されている（One Pass DP 法[20], One Stage DP 法[21]など）[27, 26]。しかし、言語的な制約を用いる場合、文脈自由文法をそのまま利用できない点や、語彙数の増大に伴う計算量の増加などが問題になる。これらの問題に対して、拡張連続 DP 法は、ワードスポッティング手法を用いてかなり少ない計算量で近似解を求めることができる[27]。また、この方法はビームサーチ法を用いることにより文脈自由文法の場合にも適用できる[27, 22]。

従来のワードスポッティング法を用いた階層型の連続音声認識システム SPOJUS-SYNO-I/II は、可能性の高い単語候補（ワードラティス）を使って文認識を行うため、文候補の探索を効率的に行えるが、ワードラティスの大きさに限界があるために認識精度が落ちることが予想された。ここで述べる方法は、従来のワードラティスを介する文認識法に対して、ワードラティスを用いない点が大きな違いである。また、通常のワードスポッティング法に基づいた方式と同様に、文法が複雑になっても認識のための計算量は殆ど語彙数のオーダーで済み、記憶量も比較的少なくて済むので、後述の One Pass 型アルゴリズムに基づく方法に比べても一般に効率が良い。しかし、構文的な制約のもとでの最適な結果が得られる保証が無く、認識精度の低下が予想される。そこで更に、構文的な制約を使ってワードスポッティングを行う方法を提案し、認識精度の改善を試みている。次の 3.4.1節では基本的なアルゴリズムを示し、その中に含まれているワードスポッティング法のアルゴリズムの部分の改良については 3.4.2節で述べる。

3.4.1　文脈自由文法に基づくフレーム同期型認識アルゴリズム

拡張連続 DP 法[27]の原理と同様に、音声の開始フレームから順に単語単位の始端点フリーのマッチングを行い、単語の始端フレーム付近で終る部分文と文法の制限をチェックしながら単語を接続していく方法である。文脈自由文法を用いると、途中に生成され得る部分文の数が膨大な数になるので、ビームサーチを採用する必要がある。図 3.3は、提案するアルゴリズムの処理内容を概念的に示している。フレームに同期して文仮説を求める処理は、(1) ワードスポッティング法によって単語仮説を求め、(2) 前述の構文解析法を用いて部分文仮説との接続を行ない、(3) 累積スコアの高い部分文仮説をビームサーチ幅の範囲内で“文法上のパス”と一緒に格納する、という基本処理の繰り返しからなる。このアルゴリズムの大まかな手順を以下に示す。

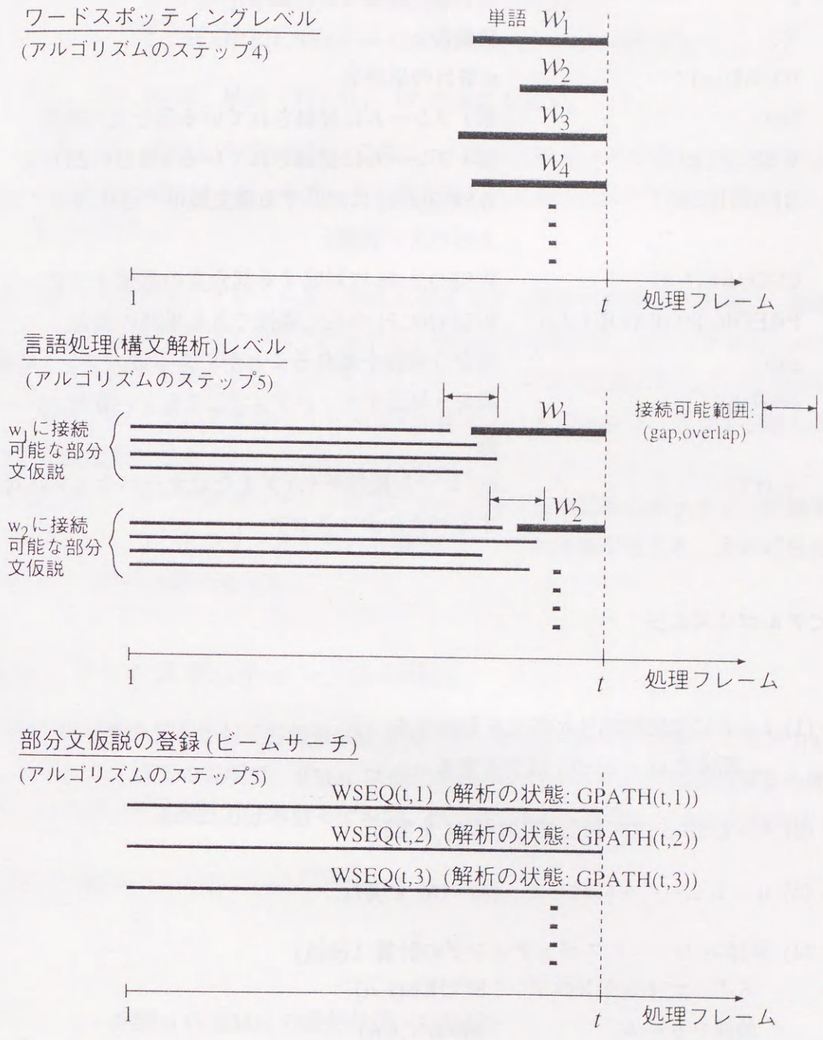


図 3.3: ワードスポッティング法に基づく統合的処理法

＜記号の定義＞

T:	入力音声の全フレーム長
N:	語彙数
WORD(n)：	n 番目の単語名
M(t)：	第 t フレームに登録されている部分文の総数
WSEQ(t,k)：	第 t フレームに登録されている k番目の部分文
GPATH(t,k)：	WSEQ(t,k) に対応する構文規則の適用履歴（文法上のパス／状態）
CSCORE(t,k)：	WSEQ(t,k) に対応する部分文の累積スコア
PREDICT(GPATH(t,k))：	WSEQ(t,k) の右に隣接できる単語の集合
gap：	隣合う単語が離れることができる最大フレーム長
overlap：	隣合う単語がオーバラップしてもよい最大フレーム長
α (r)：	rフレーム長のギャップまたはオーバラップに対応するペナルティスコア

＜アルゴリズム＞

(1) $t=0$ に文開始記号と空文を登録する.
　　部分文 ($t=1$〜T) は空とする.

(2) $t=1,2,\cdots,T$に対して (3)〜(5) を実行.

(3) $n=1,2,\cdots,N$に対して、(4)〜(5) を実行.

(4) 単語 n のワードスポッティングの計算（後述）
　　スポッティングスコア　：SCORE(t,n)
　　始端フレーム　　：BEGIN(t,n)

(5) 文仮説の生成
　　$r=-\text{gap},\cdots,-1,0,+1,\cdots,\text{overlap}$ について、(i)〜(ii) を実行

　　(i) 部分文との接続条件（ギャップ、オーバラップ）の仮定（境界フレーム）
　　b =BEGIN(t,n)+r

　　(ii) $k=1,2,\cdots,\text{M}(b)$ について、
　　$n \in$ PREDICT（GPATH(b,k)）のとき、

$$\begin{cases} \text{M}(t) \leftarrow \text{M}(t)+1 \\ \text{WSEQ}(t,\text{M}(t)) \leftarrow \text{WSEQ}(b,k)\cdot\text{WORD}(n) \\ \text{CSCORE}(t,\text{M}(t)) \leftarrow \text{CSCORE}(b,k)+\text{SCORE}(t,n)+\alpha\ (r) \end{cases}$$

　　但し、W_1·W_2は、単語（列）W_1、W_2の連結を表す。

　　このとき、文法上のパスが同じで異なった部分文はスコアが最大のものだけを残す。M(t)>BEAM（ビームサーチの最大幅）のときは、スコアが高い順にBEAM個だけ残す。

各単語のスコアは、フレーム長に対応したスコア（累積対数確率）を用い、単語列のスコアはそれらのスコアの和で評価する。単語結合の際には、ギャップやオーバラップに応じてスコアを補正する。このアルゴリズムで入力音声区間に対しての処理が終ったら、最終フレームの文仮説のうち単語列が構文的に受理可能で、スコアが最も高いものが認識結果となる。

この方法は、ワードスポッティングでフレーム毎に全単語のスポッティング結果が得られるので、ワードラティスを用いる場合に比べて処理量が増える。その対処法については、後の3.6節で述べる。

3.4.2　ワードスポッティング法の検討

実験では音響モデルとして音節HMMを使うが、単語辞書により連結して単語HMMとして用いる。以下の説明で、HMMは自己遷移ループを持たない継続時間長の離散確率分布を用いた連続出力分布型モデルで、出力確率等は対数化してあると仮定する。

＜記号の定義＞

J^n:	単語 n のHMMの最終状態（状態数）
a_{ij}:	HMMの状態 i から状態 jに遷移する確率
$b_{ij}(o_t)$：	HMMの状態 i から状態 jに遷移するとき、観測ベクトル o_tを出力する確率
$d_{ij}(\tau)$：	HMMの状態 i から状態 jに遷移するとき、継続時間長が τ である確率

$P^n(t,i)$：単語 n の HMM 状態 i，時刻 t までの最適状態遷移系列の累積対数確率（フレーム長で正規化したもの。連続 DP 法によるワードスポッティングで使用）

$L^n(t,i)$：構文的制限のない、または近似的に制限があるときの単語 HMM 系列の最後尾単語が n のときの HMM 状態 i，時刻 t までの最適状態遷移系列の累積対数確率（O(n)DP 法又は Bundle サーチ型のワードスポッティングで使用）

$B^n(t,i)$：単語 n の HMM 状態 i，時刻 t に対応する最適パスのバックポインタ（単語 n の始端フレーム−1）

(a) 連続 DP 法によるワードスポッティング

前節のアルゴリズムにおいて、拡張連続 DP 法と同様に連続 DP 法を用いる場合について述べる。HMM で連続 DP 法と同様に実現する場合は、図 3.4に示すように各単語独立に Viterbi 法による照合を行なう。連続 DP 法に基づくワードスポッティングの処理法を以下に示す[22, 39]。（HMM 構造は、非分岐型 HMM 構造、または初期状態から最終状態までの状態数の等しい分岐型 HMM とする）

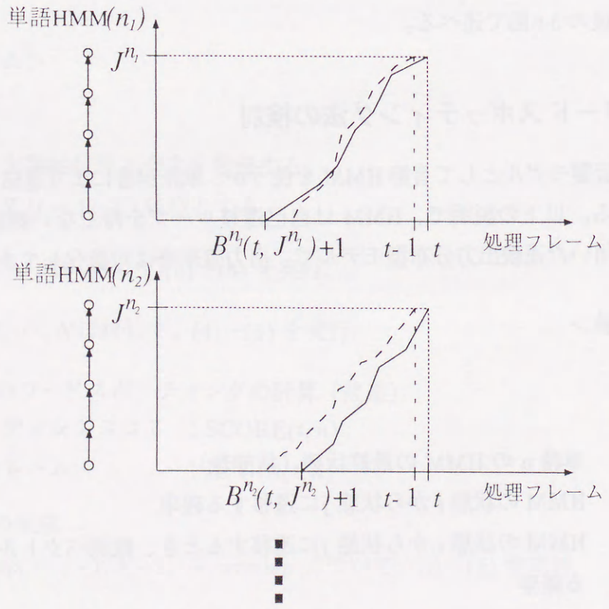


図 3.4: 連続 DP 法に基づくワードスポッティング処理

＜単語レベルのフレーム同期処理＞

$n = 1, 2, \cdots, N$について以下の処理を行う。

- 単語境界を仮定した設定

$$P^n(t-1,1) = 0 \tag{3.1}$$

$$B^n(t-1,1) = t-1 \tag{3.2}$$

- 累積対数尤度とバックポインタの計算（$i = 1, 2, \cdots, J^n$）

$$\begin{aligned} P^n(t,i) &= P^n(t-\tau', j') + \log a_{j'i} + \log d_{j'i}(\tau') \\ &\quad + \Big(\sum_{n=1}^{\tau'} \log b_{j'i}(o_{t+1-n})\Big) \Big/ \tau' \end{aligned} \tag{3.3}$$

$$B^n(t,i) = B^n(t-\tau', j') \tag{3.4}$$

但し、

$$\begin{aligned} j', \tau' &= \mathrm{argmax}_{j,\ \tau \le t}\{P^n(t-\tau, j) + \log a_{ji} \\ &\quad + \log d_{ji}(\tau) + \Big(\sum_{n=1}^{\tau} \log b_{ji}(o_{t+1-n})\Big) \Big/ \tau\ \} \end{aligned} \tag{3.5}$$

- スポッティング結果

$$\mathrm{BEGIN}(t,n) = B^n(t, J^n) + 1 \tag{3.6}$$

$$\mathrm{SCORE}(t,n) = \frac{P^n(t, J^n)}{(\text{単語 } n \text{ の音節数})} \times (t - \mathrm{BEGIN}(t,n) + 1) \tag{3.7}$$

ここで、スポッティングスコア（累積対数確率）は単語 HMM（音節 HMM の連結）の音節数に依存しなくなるように音節数で正規化して用いる。実際に前述のアルゴリズムの中で用いるときは、これまでと同じくスポッティングスコアのしきい値制限によって単語候補数を減らせる。また、フレーム同期で文レベルの認識を同時に行うので、構文的な予測が為されている単語についてのみ上述の計算をすればよく、更に計算量を減らすことができる（厳密には、単語接続時にオーバラップを考慮する場合、予測される以前のフレームが単語の始端フレームになる場合を無視している）。

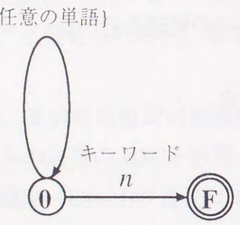


図 3.5: ワードスポッティングの音声照合における単語列の制約

(b) O(n)DP 法によるワードスポッティング

(a) で述べた方法は、単語毎に独立して計算できるため、並列計算が容易な点などで有利であるが、スポッティングスコアが単語の照合区間長に依存したスコアでは無く、スコアの正規化がヒューリスティックで単語検出の区間誤りが起こり易い問題もある。このような問題に対処するために、最適な単語系列を求める連続音声認識アルゴリズムを用いる。最適な単語系列を求める方法としては、前述の方法と計算量がほとんど変わらない O(n)DP 法がある[37]。各単語の単語境界のための初期値として、その時点までの最適単語系列の累積対数確率を用いる。終端フレーム t での単語 n のワードスポッティングは、任意の単語の連結と抽出したい単語 n を連結した単語列（図 3.5 を参照）と入力音声を照合し、単語 n の始端位置を検出することにより行なう。なお、この方法ではキーワードに先行する部分でのみ任意の単語列を許しているが、厳密にはキーワードに後続する任意の単語列も許し、発話全体の尤度を求めるほうが、スポッティングスコアの閾値の設定は容易といえる[33]。O(n)DP 法に基づくワードスポッティングのスコアは、以下の手順で求められる[32]。（本来のアルゴリズムでは、式に含まれる定数 C_p は零である）

＜単語レベルのフレーム同期処理＞

$n = 1, 2, \cdots, N$ について以下の処理を行う。

- 単語境界を仮定した設定

$$L^n(t-1, 1) = L(t-1) \tag{3.8}$$

$$B^n(t-1, 1) = t-1 \tag{3.9}$$

但し、

$$L(t) = \max_n L^n(t, J^n)$$

- 累積対数尤度とバックポインタの計算（$i = 1, 2, \cdots, J^n$）

$$\begin{aligned} L^n(t, i) &= L^n(t-\tau', j') + \log a_{j'i} + \log d_{j'i}(\tau') \\ &\quad + \sum_{n=1}^{\tau'} (\log b_{j'i}(o_{t+1-n}) + C_p) \end{aligned} \tag{3.10}$$

$$B^n(t, i) = B^n(t-\tau', j') \tag{3.11}$$

但し、

$$\begin{aligned} j', \tau' &= \mathrm{argmax}_{j,\ \tau \le t} \{ L^n(t-\tau, j) + \log a_{ji} + \log d_{ji}(\tau) \\ &\quad + \sum_{n=1}^{\tau} (\log b_{ji}(o_{t+1-n}) + C_p) \} \end{aligned} \tag{3.12}$$

C_p: 正しい単語系列の最適累積尤度に近付ける補正値（1 フレーム平均）

- スポッティング結果

$$\mathrm{BEGIN}(t, n) = B^n(t, J^n) + 1 \tag{3.13}$$

$$\mathrm{SCORE}(t, n) = L^n(t, J^n) - L(B^n(t, J^n)) \tag{3.14}$$

この方法は、連続 DP 法と異なりスコアはスポッティングされた区間長に応じたスコア体系となり、検出される単語はそれ以前の音声区間の照合結果の影響を受ける。O(n)DP 法では、構文的制約が無いため最適単語系列の累積尤度は入力音声の正しい単語系列の累積尤度以上になる。その結果として、スポッティングされる単語は正しい区間長よりも短い区間で検出され易くなるので、実験では上記の式のように最適単語系列のスコアに小さなペナルティ値 C_p を加えた。

(c) 統語的な予測を用いたワードスポッティング法（Bundle サーチ型）

上述の認識アルゴリズムはフレーム同期で文レベルの認識も行うので、各フレームにおける部分文からの統語的な予測の結果をワードスポッティングのレベルで用いることができる。そこで、前の (b) のアルゴリズムの単語境界を仮定した設定のところを変更し、それぞれの単語に対して、その単語を予測する部分文の中で最も高い累積対数確率を用いるようにする。すなわち、終端フレーム t での単語 n のワードスポッティングは、単語 n を右端に連結できる部分文と単語 n の連結の制約の下で入力音声

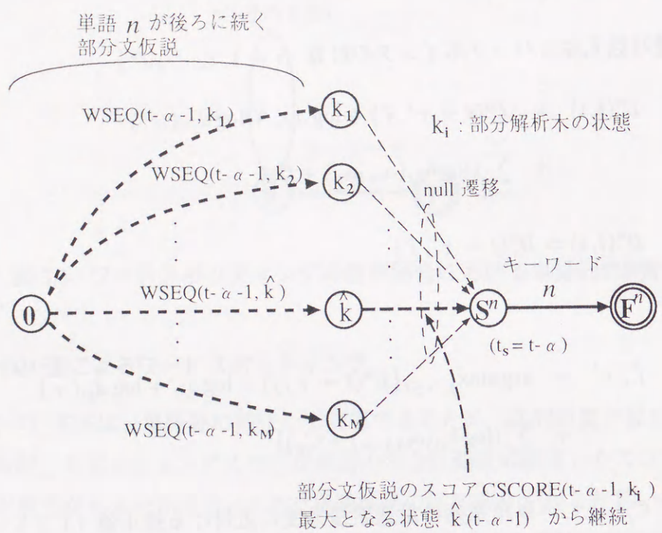


図 3.6: キーワード（検出単語）の始端フレームを $t_s = t - \alpha$ と仮定した時の部分文仮説とキーワードの接続条件

と照合し、単語 n の始端位置を検出することにより行なう（図 3.6を参照）。この場合の処理手順を以下に示す。

＜単語レベルのフレーム同期処理＞

$n = 1, 2, \cdots, N$ について以下の処理を行う。

- 単語境界を仮定した設定

$$L^n(t-1,1) = \max_k \mathrm{CSCORE}(t-1,k) \tag{3.15}$$

$$B^n(t-1,1) = t-1 \tag{3.16}$$

但し、$n \in \mathrm{PREDICT}(\mathrm{GPATH}(t-1,k))$

- 累積対数尤度とバックポインタの計算

前述の O(n)DP 法での計算内容 (式 (3.10),(3.11)) と同じ

- スポッティング結果

$$\mathrm{BEGIN}(t,n) = B^n(t, J^n) + 1 \tag{3.17}$$

$$\mathrm{SCORE}(t,n) = L^n(t, J^n) - L^n(B^n(t, J^n), 1) \tag{3.18}$$

前の二つと同じく、単語毎に一つのスポッティング結果（スコア、区間）が計算される。実際には、ある単語は幾つかの異なる部分文から予測されており、各々の部分文との接続を仮定して独立に最適なパスを計算するべきであるが、上述のアルゴリズムではそれらの部分文の最大スコアを初期値として近似している。しかし、前節のアルゴリズムに示したとおり、単語のスポッティング結果は、その単語が接続可能な部分文全てと接続を行って文レベル処理を行っているため、結果的に同一単語に対する最適パスの探索を一つにまとめていると見ることができる。すなわち、これは渡辺らが提案している「Bundle サーチ」[34]（オートマトン制御の連続音声認識アルゴリズムでの適用）と同様な原理である。ここで述べる方法はビームサーチ及び文脈自由文法を用いる場合の適用法と考えることができる。さらに部分文と単語との接続時にギャップやオーバラップを考慮する点は、この近似化された手法の認識精度の低下を抑える点で有効に働くものと考えられる。

3.5 One Pass 型アルゴリズムに基づいた統合化 - SPOJUS-SYNO X -

One Pass DP 法[20, 21]は、有限状態オートマトン（ネットワーク）による構文制御が可能で効率的な認識法であるが、本システムのように文脈自由文法で記述された構文規則をそのまま扱うことはできない。そこで、3.3.2節で述べたように、構文解析法による部分文の仮説の予測によって、有限状態オートマトンで表現される構文情報を動的に拡張する。このとき、状態（部分文）単位のビームサーチ法を使って、現在処理対象となっている文脈自由文法の一部から有限状態オートマトンへの動的な展開を行い、その結果を構文制御に用いる。

3.5.1 有限状態オートマトンの動的な展開

フレーム同期型の認識アルゴリズムでは、複数の部分文仮説に対する音声の照合区間が共通のため、仮説の尤もらしさの比較が容易であり、ビームサーチ法を容易に適用することができる。例えば、ある部分文仮説に対する尤度がしきい値以上である場合だけ、部分文の後ろに隣接する単語の予測を行なって仮説を展開すれば、音声照合スコアに基づいて探索空間を動的に制限することができる。しかし、このような方法

を適用しても、入力音声に対して生成される仮説の数は非常に多くなるため、文法によるトップダウンの単語予測を行なうための計算量は無視できない。特に、文脈自由文法による解析では、部分文に対する文法のあいまいさがあるために、同じ解析の状態を持つ部分文仮説が複数生成され、それらの仮説に対しての予測の処理が重複して繰り返される問題が生じる。このような問題に対して、LR パーザを用いて予測を行なう場合は、解析表の状態を考慮することで回避することができる。ここでは、3.3.2節で述べた Earley のトップダウン型構文解析法を用いる場合の、効率的な予測処理の実現について述べる。

3.3.2節で述べた Earley 法に基づく構文解析法は、解析の状態を文法上のパスで記憶することによって効率的な予測を実現するものであった。更に上述のような重複した予測を回避するために、ここでは予測の結果を有限状態オートマトンの形で記憶することを考える。そのためには、前述の構文解析法による予測で新たな文法上のパスが得られるたびに、その文法上のパスと一対一で対応するオートマトンの状態へ対応付ければよい。

図3.2の文法の例で説明すると、まずオートマトンの初期状態 q_0 は文法の開始記号の文法上のパスに対応付けられる。すなわち、

$$q_0 = \text{“5”}. \tag{3.19}$$

初めは、この初期状態だけを予測の候補として文頭の単語を予測し、次のように新たな文法上のパス及び単語が得られる。

文法上のパス	予測された単語
“6 11”	A, THE
“6 16 21”	BIG, YOUNG
“6 16 26”	I, JOHN, MAN, TENNIS

ここで得られた3種類の文法上のパスを、新たにオートマトンの状態とする。すなわち、この時点までのオートマトンの状態集合 $\boldsymbol{Q}$ は次のようになる。

$$\boldsymbol{Q} = \{\{\text{“5”}\}, \{\text{“6 11”}\}, \{\text{“6 16 21”}\}, \{\text{“6 16 26”}\}\} \tag{3.20}$$

図3.7は、前述のようにビームサーチ法を適用し、尤もらしい複数の部分文仮説（実際には、それらの仮説に対する文法上のパス）からの予測をもとにオートマトンの形へ展開していく過程の概要を示している。この例では、状態 p（に属する複数の部分文仮説）の尤度がしきい値以上で予測の候補となっている場合を想定しており、その状態に対応する文法上のパス “7 37 11”（図3.2の文法において、例えば “I know a” のような部分文仮説に対応する。）に基づいて次単語を予測し、その結果得られた新たな文法上

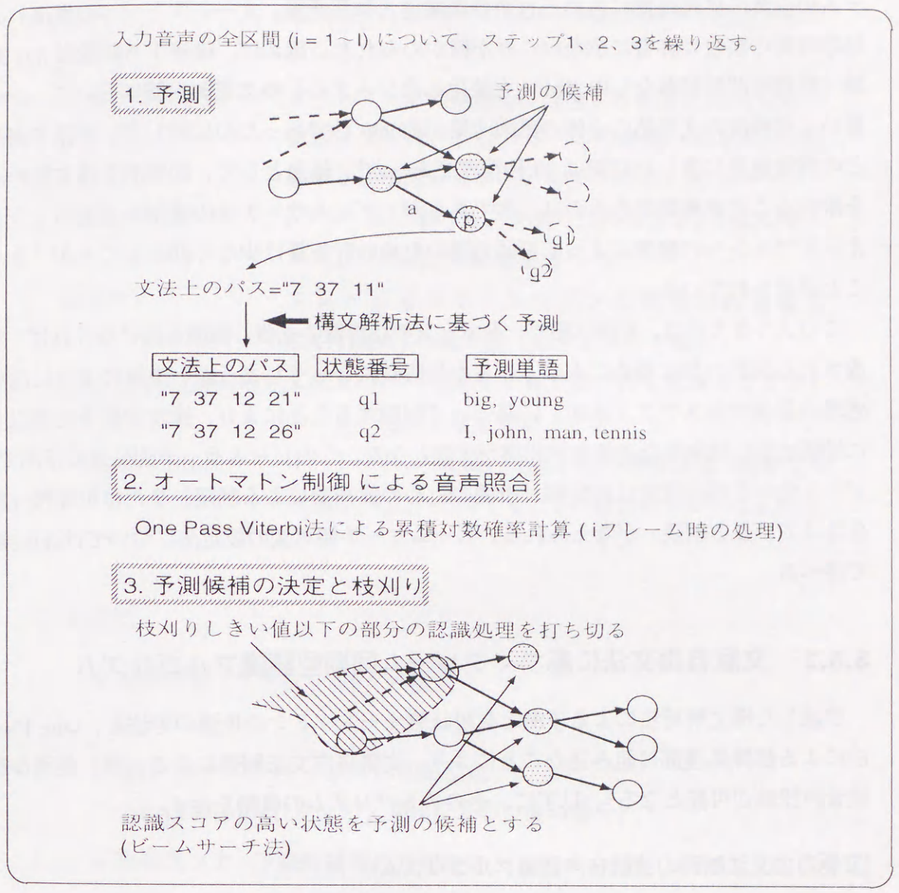


図 3.7: フレーム同期型の認識処理における有限状態オートマトンの展開の過程

のパスをオートマトンの状態 q_1, q_2 に対応付けている。このとき、予測された単語は、オートマトンの状態 p から q_1 または q_2 の間のアークに対応付ける形で記憶される。

このようにして展開されたオートマトンの構文制約の下で、後述のアルゴリズムで音声照合を進めていく（図3.7の第2ステップ）。処理フレームが進むうちに、定められたスコア規準を満たした部分文仮説が出現した場合には、それらの仮説に対応するオートマトンの状態を新たな予測のための候補とする（図3.7の第3ステップ）。

新たな予測のための候補が現れた場合には、上述のようにまた予測を行ない、新たな文法上のパス及び単語を求め、オートマトンを拡張する（図3.7の第1ステップ）。

このようにオートマトンへの動的な展開を行なう場合、全体的に必要となる処理量の大きさが重要な問題である。しかし、後述の認識タスクによる評価では、認識シス

テムの全体の処理時間に占める言語処理関連（単語予測、オートマトンへの展開）の処理時間の割合は非常に少ないことが確かめられた。例えば、後述する語彙数521単語（継続時間長制御なしのHMMを使用したシステム）の文認識実験において、やや長い4秒程度の文発話の全体の認識時間が約980秒であったのに対して、単語予測などの言語処理に要した時間は20秒程度であった[4]。結果として、効率的な構文解析法を用いることが重要であるのは当然であるが、ビームサーチ法の適用と上述のようなオートマトンへの展開によって言語処理のための計算量は少なく抑えることができることが示されている。

このような方法は、有限状態オートマトンの状態数の拡張に制限を設けなければ、生成される状態の数は極めて大きくなるか無限数になってしまうが、上述のように認識処理の累積照合スコア（確率）に基づいて制限することにより、探索空間を必要以上に拡張せずに効率的な連続音声認識が可能となる。このビームサーチ法による予測で、ビームサーチ幅の設定は固定幅（予測のための候補数による制限）または可変幅（照合スコアによる制限）が考えられる。ビームサーチ幅可変の設定法については3.6.2節で述べる。

3.5.2　文脈自由文法に基づくフレーム同期型認識アルゴリズム

前述した構文解析法による予測と有限状態オートマトンの拡張の方法を、One Pass法による認識処理部に組み込むことにより、文脈自由文法制御による（準）最適な連続音声認識が可能となる[5]。以下に、そのアルゴリズムの概略を示す。

[文脈自由文法制御の連続音声認識アルゴリズム]

I	入力フレーム長
Q	生成されているオートマトンの状態数
$\mathbf{GPATH}(q)$	オートマトンの状態 qに対応する文法上のパス
J^n	単語 n の HMM の状態数
$P_q(i)$	i フレームにオートマトンの状態 qに到達するときの最適パスの累積対数確率
$N_q(i)$	i フレームにオートマトンの状態 qに到達するときの最適パス上の最終単語

[4]OMRON 社製ワークステーション LUNA88k(25MIPS) での CPU 時間の実測値。
[5]ここで（準）最適であるのはビームサーチ法を採用しているためである。

$B_q(i)$	単語 $N_q(i)$ に対応するバックポインタ（始端フレーム－1）
$p_q^n(i,j)$, $b_q^n(i,j)$	オートマトンの状態 qに到達する単語 n の HMM の状態 j までの、1〜i フレームの最適パスの累積対数確率と、単語 n に対するバックポインタ
$\mathbf{CANDS}(i)$	i フレームにおける、予測のためのスコアの高いオートマトンの状態の候補集合
BEAM	フレーム毎のオートマトンの状態の候補集合 $\mathbf{CANDS}(i)$ の最大数（ビームサーチ幅）
$\mathbf{PREDICT}(gpath)$	文法上のパス $gpath$ から予測される単語及び文法上のパスの対からなる集合
$\mathbf{FAtable}(s_{dest})$	オートマトンの状態（s_{dest}）毎の遷移情報（遷移元の状態と、s_{dest}へ遷移するときの単語の対からなる集合）

1. 初期化

   - 初期状態（状態番号= 0）の設定：
     $Q = 0$.
   - 文法上のパスのテーブルの初期設定：
     $\mathbf{GPATH}(0)$="(CFG の文開始記号の位置の番号)".
   - 照合スコア、予測候補等の初期設定：
     $P_0(0) = \log(1.0), \quad B_0(0) = 0,$
     $P_0(i) = -\infty \ \ (i = 1, 2, \cdots, I),$
     $\mathbf{CANDS}(0)=\{0\}$.

2. $i = 1, 2, \cdots, I$.について、3〜6 を実行

3. 構文解析による予測とオートマトン遷移情報の更新
   オートマトンの状態 $s_{orig} \in \mathbf{CANDS}(i-1)$ に対して、(1)〜(3) を実行。但し、状態 s_{orig}に対して既に実行していれば省略。

   (1) 状態 s_{orig}の次に来る単語を予測し、予測された単語 n とその時の文法上のパス gの対 (n, g) を全て含む予測結果の集合を求める。
   $\mathbf{PREDICT}(\mathbf{GPATH}(s_{orig})) = \{(n_1, g_1), (n_2, g_2), \cdots, (n_M, g_M)\}$
   但し、M はオートマトンの状態 s_{orig}から予測された文法上のパスの数

(2) (1) で得られる文法上のパス $gpath \in \{g_1, g_2, \cdots, g_M\}$ について、それぞれのオートマトンの状態番号 s_{dest} を次のように求める。

$gpath \in \mathbf{GPATH}(q)$ $(q = 1, \cdots, Q)$ の場合：…(状態番号を割り当て済み)

$s_{dest} \leftarrow q$ （但し、qは次式を満たすもの。$gpath = \mathbf{GPATH}(q)$）

$gpath \notin \mathbf{GPATH}(q)$ $(q = 1, \cdots, Q)$ の場合：… (新しい状態番号の割り当て)

$Q \leftarrow Q + 1$

$\mathbf{GPATH}(Q) \leftarrow gpath$

$s_{dest} \leftarrow Q$

(3) (2) で得られる遷移先の各状態 s_{dest} 毎に、前状態 s_{orig} と予測された単語 n の対 (s_{orig}, n) を $\mathbf{FAtable}(s_{dest})$ に追加する。

4. One-Pass search アルゴリズム（オートマトン制御、ビームサーチ）

(1) $q = 1, 2, \cdots, Q$ について、(2)～(5) を実行

(2) 各単語 n $(\{n; (s, n) \in \mathbf{FAtable}(q)\})$ について、(3)～(4) を実行。但し、枝刈りのためのマークが (q, n) に対して付いているときは、$p_q^n(i, J^n) = -\infty$ として、計算を省略する。

(3) $p_q^n(i, j) = -\infty$　　for　$j = 1, 2, \cdots, J^n$

$\max_s P_s(i-1) > p_q^n(i-1, 1)$ のとき、

$$\begin{cases} p_q^n(i-1, 1) = \max_s P_s(i-1) \\ b_q^n(i-1, 1) = i - 1 \end{cases}$$

但し、$(s, n) \in \mathbf{FAtable}(q)$　…… (*注：後述)

(4) $j = 1, 2, \cdots, J^n$ について、$p_q^n(i, j)$ および $b_q^n(i, j)$ の更新　（Viterbi アルゴリズム、文献[39]参照）

(5) $\hat{n} = \mathrm{argmax}_n p_q^n(i, J^n)$

$P_q(i) = p_q^{\hat{n}}(i, J^{\hat{n}})$

$B_q(i) = b_q^{\hat{n}}(i, J^{\hat{n}})$

$N_q(i) = \hat{n}$

5. 枝刈り判定（3.6.2節参照）

$\max_j p_q^n(i, j)$ の値によって、しきい値以下の (q, n) の組にマークを付ける

6. 予測のためのオートマトンの状態の候補リスト作成

$$\mathbf{CANDS}(i) = \bigcup_{1 \le n \le BEAM} \text{arg-}n\text{th-}\max_q P_q(i)$$

ここで、候補リストに含まれる状態 q と任意の単語 n との組 (q, n) に枝刈りのマークが付けらているものはマークを取り消す。

7. 認識結果の出力　（$P_q(i)$, $B_q(i)$, $N_q(i)$ によるバックトレース処理, 文献[39]参照）

上記アルゴリズムでは、各処理フレームで全ての部分文仮説[6]から、既に照合が開始されている後続単語への接続を仮定している。しかし一般には、フレーム同期のビームサーチ法としては、各処理フレームで一定個数（あるいはスコアがしきい値以上）の部分文仮説からだけ後続単語への接続を仮定する。その場合には、上記アルゴリズムの第4ステップの (*) の条件を次のように変更する。

但し、$(s, n) \in \mathbf{FAtable}(q)$ かつ、$s \in \mathbf{CANDS}(i-1)$　…… (**)

(*) の条件を用いた場合はビームサーチの制限が緩くなるため、ビームサーチ幅を大きくするのと効果的にほぼ同様といえる。但し、(*) の条件では、一旦照合が開始された仮説については枝刈りされるまで各フレームでの単語接続を仮定した処理を行なうので、単純にビームサーチ幅を増やして (**) の条件を使った場合に比べて、最適解がより保証され易いといえる。しかし、各仮説の照合の打ち切りをスコアの基準などによって判定する場合、打ち切りの判定が遅れ易くなって全体的な計算量が増加することも考えられる。後述の実験と同様に語彙数 500 単語のタスクで認識実験を行なった結果では、これらの条件の違いによる処理量や認識精度の差はなかった（一フレーム平均の照合単語数の差は 3%程度で、処理時間の差は測定誤差内）。しかし、後の章で述べる未知語処理を併用した場合（最大ビームサーチ幅が 20）の認識実験結果では、未知語（言い直し）を含む認識結果の一部で、(*) の条件の方だけでより最適な結果が得られる場合があった（認識率で 1%程度の差）。予想される原因の一つとして、言い直しを含むような発話では一部で発音があいまいになりがちであり、展開される仮説数も増大しやすいため、“未知語” を含んだ最適な仮説が洩れやすい、ということが考えられる。

3.6　フレーム同期アルゴリズムにおける高速化

3.6.1　ワードスポッティング法ベース手法の計算量の削減

3.4節で述べた方法において、文レベル及び単語レベルの処理の計算量を削減する方法を考える。このアルゴリズムでは、各フレームの処理において、スポッティングされ

[6] HMM による照合では DP マッチングにおける傾斜制限のようなものが一般にはないので、照合を打ち切る判定をしない限り処理が継続されて仮説が残されている。

た単語と部分文の接続を構文的に制限する。そこで、ワードスポッティングの処理の他に、単語の予測や単語列の生成、文仮説スコアの計算とソーティング（ビームサーチのため）、などを考慮しなければならない。

単語の予測に関しては、3.4.1節で説明したフレーム同期のアルゴリズムの場合、あるフレームの処理において部分文との接続点が単語毎に異なるため、全フレームについて（ビームサーチ幅の）部分文からの単語予測の結果を保持しておく必要がある。単語の接続毎にパーザで予測すれば記憶量は節約できるが、計算回数が非常に増えてしまう。この問題に対処するため、従来の文認識法を改良し、単語予測の度に求まる文法上のパス（構文規則の適用履歴）を3.5.1節で述べた方法によって有限状態オートマトンの状態と対応づけて、オートマトン文法の形で予測結果を保持するようにしている。これによって、重複して構文的な予測を行うことは無くなり、処理中に部分文を保持する時も、文法上のパスに対応する状態番号を残しておくだけでよい。

上述した方法により、単語の予測に関してはそれほど計算量の問題はなく（全体の計算量の数%程度）、むしろビームサーチに関連する単語列の生成や文スコアの比較などの計算回数が問題である。これらの計算は、各フレームで得られる単語候補の数や、ビームサーチ幅、2単語間の接続時の許容範囲（gap,overlap）によって、ほぼ決まる。各フレームでの単語候補の数は、3.4.2 節で触れたように部分文から予測されている単語のみ扱うようにすることで全体的に半分程度の単語数まで減らせることが実験で確認された。また、3.4.2節の (b),(c) の方法のように音声の始端フレームからの累積対数確率を計算するワードスポッティング法であれば、ある処理フレームで最大の累積対数確率との差をとって単語候補をしきい値で制限したり、フレーム単位でワードスポッティング処理を中断することができる[35, 36]。これで、文レベル処理のみでなく単語レベル処理の計算量も削減され、文認識時間の短縮にかなり効果がある。iフレームでの単語 n に対するワードスポッティング処理の中断は、次式の累積対数確率の差で判断する。

$$\max_j L^n(i,j) < L_{max}(i) - \lambda(i) \tag{3.21}$$

但し、

$\lambda(i)$：枝刈り判定用のしきい値

$L_{max}(i) = \max_{j,n} L^n(i,j)$

単語候補の制限は、上式で左辺を $\log L^n(i,J^n)$ に置き換えて同様に判定する。

3.6.2 One Pass 法のビームサーチを用いた高速化

これまで述べてきた構文制御に基づく連続音声認識アルゴリズムは、フレーム同期的に認識処理の探索空間を広げていくため比較的効率的な方法である。しかし、構文解析による予測でオートマトンの状態数が増加するにつれてフレーム毎の計算量も増えていくので、何らかの処理削減を行ったほうがよい。

そこで、One Pass 法での Viterbi サーチにおいてビームサーチを導入する。DP マッチングの場合は沢井ら[35]、迫江ら[36]が DP パスにおけるビームサーチの導入を提案している。HMM の場合、Viterbi パスにおけるビームサーチは、各フレームで一定個数からの後続状態を予測して残していくだけの場合には、状態数が比較的少ないことを考えると効果はそれほど大きくないものと予想される[7]。また、構文知識を用いた文認識の場合、HMM の状態単位のビームサーチは展開される部分文仮説が多くなるにつれてオーバヘッドが増加する。そこで、高速化処理の負担を軽くすることも考えて、累積対数確率の漸化式計算の中で、特にオートマトン上の各状態に遷移する単語単位のレベル（すなわち、$p_q^n(i,j)$ 計算の状態 q 及び単語 n の組合せ単位）で枝刈りを行う。枝刈りの判定を行う処理は、前述の認識アルゴリズムの第5ステップでフレーム毎に行われるが、その具体的な判定法（i フレーム時）を以下に示す。

1) 前述（3.5.2節）のアルゴリズムの第4ステップで求まる累積対数確率 $p_q^n(i,j)$ の、i フレームにおける最大値 $P_{max}(i)$ を次のように求める。

$$P_{max}(i) = \max_{j,n,q} p_q^n(i,j) \tag{3.22}$$

2) 各状態 q の各単語 n に対して次式の条件をチェックし、これを満たすとき枝刈りを行う。

$$\max_j p_q^n(i,j) < P_{max}(i) - \lambda(i) \tag{3.23}$$

但し、$\lambda(i)$：枝刈り判定用のしきい値

枝刈り判定用のしきい値 $\lambda(i)$ は、あらかじめトレーニング用文データに対して最適照合パスを求め、その時の $P_{max}(i) - \max_j p_q^n(i,j)$ の値を調べることによって、適当な値を決めることができる。参考のために、3.8節の「電子メール」タスク実験で得られた、テスト用文データの全ての正しい認識結果に対しての $P_{max}(i) - \max_j p_q^n(i,j)$ の値のプロットを、図3.8に示す（点線は認識実験時のしきい値設定の例）。

[7] 標準的な HMM で継続時間の制限を与えない場合は、文末まで照合が続行されるため。

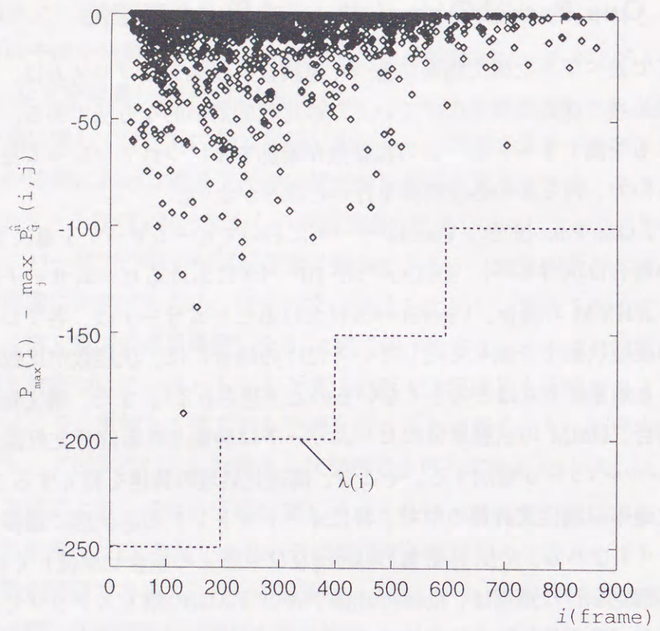


図 3.8: 累積対数尤度の最大値と最適パス上の累積対数尤度との差の分布

ここで注意を要するのは、一般にオートマトンによる構文制約のネットワークはグラフ状になっているが、認識アルゴリズムはフレーム同期型であるということである。つまり、ある時点 i で状態 q、単語 n の最適パスに対応する部分文仮説の尤度が式 (3.23) の条件を満たして枝刈りされても、後のある時点では状態 q、単語 n の先行部分の最適な部分文仮説が変わり、再び予測されることがあるということである。従って、状態 q、単語 n に対する照合スコアの計算が後に再開され得ることを考慮しておく必要がある（3.5.2節のアルゴリズムの第4,5,6 ステップを参照）。

この枝刈りの方法は、前述のアルゴリズム（第 6 ステップ）で、予測のためのビームサーチ幅を制限する（可変にする）のにも適用できる。予測はフレーム毎に行われるので、単語のフレーム長を考慮してもビームサーチ幅が可変であった方が効率的である。そこで、ビームサーチの予測の候補となる各状態 qを、次式の累積対数確率の差で制限する。

$$P_q(i) \geq P_{max}(i) - \lambda(i) \tag{3.24}$$

$\lambda(i)$ の値は、前の枝刈り時の値と同じである。これにより、予測のためのビームサーチ幅は、最大幅（固定幅の値）の範囲内で、フレーム毎に可変の値となる。この枝刈り、

予測制限のしきい値$\lambda(i)$ を適当に定めることによりにより、認識精度を落とす事なく余分な認識照合を防ぐことができ、認識時間の短縮および記憶量の節約を期待できる。

3.6.3 単語継続時間長の制限

基本的な HMM では、各状態の継続時間に対して自己遷移ループで対応させている。しかし、自己遷移ループに対する継続時間長 t の確率は、$p_{ii}^{t-1}(1-p_{ii})$ となることから、遷移確率だけでは継続時間の情報を十分に表現することができない。そこで、本研究の実験では、2.4.3節で述べたように状態毎の離散継続時間確率分布を用いた継続時間長制御付き HMM を使用している。このモデルは後の実験で示されているように、連続音声認識の精度向上にかなり効果がある。

しかし、ある単語発音の一部分と音響的に良く似ている単語の場合、その単語の平均的な継続時間長とかなり違っていても照合スコアが高くなるような場合がある。その結果、それに続く単語が予測されてしまい、余分な照合が増加することがかなりあることが予想される。そこで、単語継続時間長の制限を設けることによって、誤った単語からの継続を排除することを試みる。但し、この目的においては、継続時間長の上限はあまり効果が期待できないので、下限値だけを設定する。

ワードスポッティング法に基づく方法では、得られる単語候補を単語のフレーム長によって制限すればよい。また One Pass 法に基づく方法では、3.5.2節のアルゴリズムの第 4 ステップで、単語の累積対数確率が計算される度に、その単語のフレーム長を $i - b_q^n(i, J^n)$ で求め、制限された長さより短ければ $p_q^n(i, J^n) = -\infty$ とすればよい。この制約により、予測時の制限に特に貢献するものと考えられる。

3.7 探索・照合を効率化するための言語処理レベルの改良

3.3.2節で述べた構文解析法を音声処理と統合した場合に、構文情報を動的に拡張していく方法において、主に二つの問題点が考えられる。

一つは、前述のように解析の状態を記憶していく方法において、得られた文法上のパスが異なる場合もその後の構文的な機能（後続可能な単語系列）が全く同じになるものがあるという問題である。例えば、文末の同一の単語は前述の One Pass(Viterbi) 法では照合をまとめることができるが、文法上のパスが異なる場合は別々に照合される。この問題の原因は、前述の方法では、ある書き換え規則で右辺の最後まで適用が終了した時点にすぐに還元を行なっていないためである。更に後続単語まで処理が進められた段階では解析の状態は同一になるが、状態が同一になるのが遅れるため、音声処理レベルの照合が一部重複して無駄に行なわれる可能性がある。

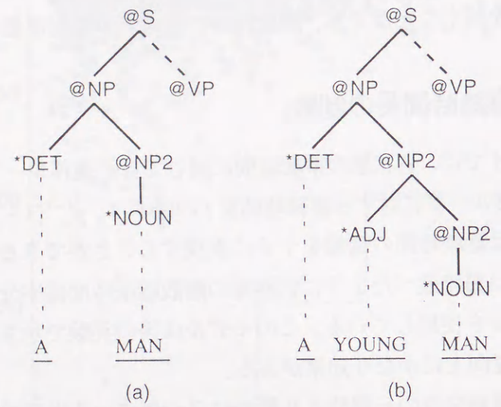


図 3.9: 部分的な構文解析木の例

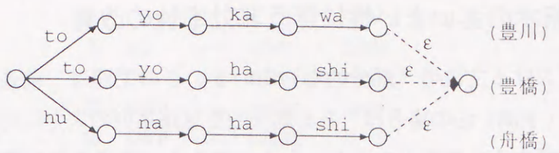
もう一つは、文法が同一の単語列に対して複数の（部分）解析木を得るような「あいまいさ」を持っている場合、前述の方法は単語列の重複した探索・照合が行なわれることがあるという問題である。この 2 つの問題点を改善するため、本研究で検討した方法について述べる。

3.7.1　文法上のパスのあいまい性の改善

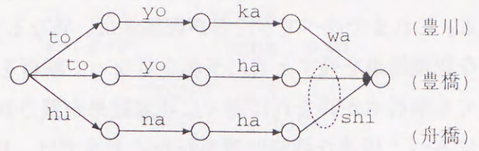
まず、図 3.2の文法の例を使って問題となる例を示す。文の途中までの生成で次の 2 つの部分文が導出されるとき、それらの部分解析木は図 3.9のように得られる。これらの解析木を見ると、二つの部分文は共にその後に非終端記号が@VP となる単語系列だけが接続できるのは明らかであり、後続可能な単語系列の集合は全く同じになることが分かる。しかし、従来はこれらの部分文（の最後の単語）が予測された時点で得られる文法上のパスは、

	部分文		文法上のパス
(a)	A MAN	:	“6 12 26”
(b)	A YOUNG MAN	:	“6 12 22 26”

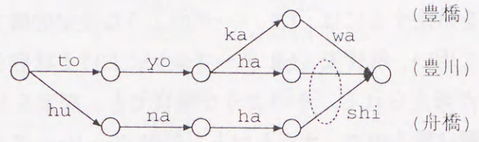
のように異なって得られる。更に一単語先まで処理された時点で初めて、同一のパスが得られる。このように異なった文法上のパスが得られる問題は、LR パーザのようなシフト還元構文解析法における還元と同様な処理を加えることで容易に解決できる。つまり、ある終端記号（ワードクラス）の予測が行われる度に、書換え規則のその終端



(a) 従来の解析（予測）法



(b) 先読みによるオートマトン生成の効率化



(c) 決定性オートマトンへの変換

図 3.10: 構文解析法により生成される単語表現オートマトン
（One Pass 法（Viterbi 法）では点線の弧の部分の照合がまとめられる。）

記号が導出された箇所の右隣りを常に先読みする。もし、右隣りに何か記号があれば文法上のパスはそのままとし、記号が存在しなければ右隣りに記号が見つかるまで適用履歴を逆戻りし（完了操作，還元）、その結果得られる文法上のパスを途中解析結果として記憶する。上述の例の場合、この改良を加えれば両者とも同じ文法上のパス “6” が得られる（両者とも@NP の完了操作が行われるため）。この改良により、展開される有限状態オートマトンはより簡潔なものとなる。

上述の効率化の過程が分かり易いように、孤立単語認識で文法の終端記号が音節や音素のような単位の場合で考える。例えば、各単語の音声表記に従い音節列からなる規則を各単語毎に列挙した場合、認識時に予測される文法上のパスをそれぞれオートマトンの状態と考えると図 3.10(a) のように表現される。上述のように簡単な先読みを行なってから文法上のパスを得ることによって、図 3.10(b) のようにオートマトンがより簡潔になり、一部の終端記号の照合がまとめられる。文レベルの文法で単語が終端記号の場合にはまとめられる単位が単語となるので、一般に処理量の削減のための効果がより大きいといえる。（なお図 3.10(c) については後述）

3.7.2 解析木のあいまい性に伴う非効率性の改善

構文解析における文法の（部分的な）あいまいさの問題は、前述のような *breadth-first*型の解析（予測）法の場合は、ある部分的な単語系列の入力に対して複数の解析木の可能性を記憶する必要性があるということである。前述の音声認識法における単語予測という立場から言えば、同じ単語が複数の部分解析木の可能性を持って予測され得るということである。これまで述べてきた音声認識法は、異なる文法上のパス（部分解析木）単位で最適な認識結果を残すと共に予測を行なって解析を進めていく。従って、同じ単語列に対しても解析木が異なれば別々に認識結果が残されていた。また、3.5節で述べた構文制約に基づく最適な認識処理を行なう方法では、同じ単語列でも解析木が異なれば別々に音声と照合される問題があった。

このような問題を解決するには、LR パーザのような決定的構文解析法を前述の予測的構文解析器として用い、解析表（LR テーブル）における状態を最適な認識結果を残す単位とする方法が考えられる。そのような場合でも、探索をトリー状に行なっていけば依然として問題は残るので、オートマトン制御 One Pass アルゴリズムのようにグラフ上での探索として扱わねばならない[28]。

これまで利用している構文解析法でこの問題を解決するには、一つの文法上のパスに対する単語予測で得られる複数の文法上のパス（部分解析木）を、予測された異なり単語毎にまとめて、まとめられた文法上のパスの集合をそれぞれ一つのオートマトンの状態とみなすとよい。この解決の方法は、Earley のアルゴリズムにおいて入力の単語を読み進める毎に求めるアイテム集合を、それぞれ一つの状態と見なすことにほぼ等しい。また、その処理の内容は、非決定性有限状態オートマトンを決定性オートマトンへ変換するのと同じ考え方に基づいている。

例を示すと、図3.11のあいまいさを持った文法において文頭の単語を予測した場合、予測される文法上のパスと単語（ここではワードクラス）は次のように得られる。

文法上のパス	予測された単語(ワードクラス)	
“6 16”	*DET	···①
“6 21 26”	*ADJ	···②
“6 21 31”	*NOUN	···③
“11 16”	*DET	···①’
“11 21 26”	*ADJ	···②’
“11 21 31”	*NOUN	···③’


```
      0      1      2      3 ...
 5  @S  →  @NP    @VP
10  @S  →  @NP    *AUX   @VP
15  @NP →  *DET   @NP2
20  @NP →  @NP2
25  @NP2 → *ADJ   @NP2
30  @NP2 → *NOUN
35  @VP →  *VERB
40  @VP →  *VERB  @NP

*NOUN → I
*NOUN → JOHN
*NOUN → MAN
*NOUN → TENNIS
*AUX  → WILL
*AUX  → CAN
*VERB → KNOW
*VERB → PLAY
*DET  → A
*DET  → THE
*ADJ  → BIG
*ADJ  → YOUNG
```

図 3.11: あいまいさを部分的に含む文脈自由文法の例

このように6種類の文法上のパスが得られるが、単語（ワードクラス）毎にまとめることによって得られる3種類の文法上のパスの集合を、それぞれオートマトンの状態に対応付ける。すなわち、この時点までのオートマトンの状態集合 Q は次のようになる。

$$Q = \{\{\text{“5”}, \text{“10”}\}, \underbrace{\{\text{“6 16”}, \text{“11 16”}\}}_{①,①'}, \underbrace{\{\text{“6 21 26”}, \text{“11 21 26”}\}}_{②,②'}, \underbrace{\{\text{“6 21 31”}, \text{“11 21 31”}\}}_{③,③'}\} \tag{3.25}$$

ここで、{“5”, “10”} は初期状態に対応するものである[8]。

前節で示した孤立単語の文法（終端記号が音節）の例を考えると、上述のように解析木のあいまい性をまとめて扱うことにより語頭の同じ音節列の部分がまとめられ、図3.10(c) のようにオートマトンが簡単化される。この例では語尾の/ha shi/の部分がまとめられないが、このような問題に対しては、3.8.9節で述べるようにあらかじめ文法をオートマトンの最簡形が得られるように変換することで解決できる。従って、単語内の規則のように解析木を得ることが重要でない場合は、認識処理のオーバヘッドを

[8]現在のシステムでは、`@S0 → @S` という様な書き換え規則を追加して初期状態に対応する文法上のパスを一つにする必要がある。

A ⟶ C D E F
A ⟶ C D F G
B ⟶ H E F
B ⟶ H F G
(a) 元の書き換え規則

A ⟶ C D X
B ⟶ H X
X ⟶ E F
X ⟶ F G
(b) 変換後の書き換え規則

図 3.12: 文脈自由文法規則のあいまい性の改善

少なくするために、あらかじめ効率的な書き換え規則に変換しておくことが望ましい。文レベルの文脈自由文法規則を記述する場合においても、あいまい性の生じ方が文法の記述の仕方に依存するという問題がある。そこで、図 3.12のように文レベルの書き換え規則を変換することによってあいまい性を減らす処理も容易に実現できる。但し、この場合、構文木が変わってしまうことに注意が必要である。

3.8 文認識実験

ここでは、本章で述べたアルゴリズムに基づくシステム (SPOJUS-SYNO·II,III,X) を用いて認識実験を行なった結果を示す。

3.8.1 システム構成

図 3.13に、SPOJUS·III/X のシステムの概略を示す。

音響処理部においては、表 3.1の条件で入力音声の分析を行う。これによって、入力音声は 10 次の LPC メルケプストラム系列に変換される。音声認識に有効とされている LPC メルケプストラム係数の回帰係数は本章の実験では用いていない。

表 3.1: 音声資料の分析条件

サンプリング周波数	12 kHz
窓関数：ハミングウィンドゥ	21.33 msec (256 points)
フレーム周期	5 msec (60 points)
分析	14 次の LPC 分析
特徴パラメータ	10 次 LPC メルケプストラム係数

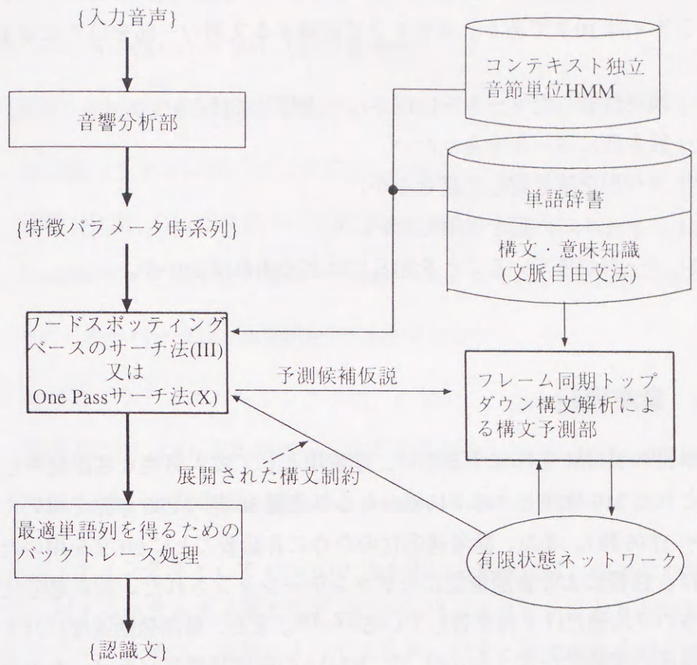


図 3.13: システム（SPOJUS-SYNO·III/X）の概略図

音声認識処理では、音響的モデルとしてコンテキスト独立の音節単位の HMM を用いる。構文解析処理が扱う構文・意味的な知識は単語単位で記述されているので、認識処理では単語辞書を参照しながら単語の照合のための音節 HMM の系列を自動的に構成する。HMM の種類としては継続時間長制御付連続出力分布 HMM（DDCHMM）[22, 39]を使用している。モデル構造はそれぞれ 5 状態 4 出力分布で、各状態毎に離散継続時間確率分布を持っている。用意された HMM は、外来語を含む語い単語を構成するために 110 個の音節に対応している。

3.8.2 タスク

認識対象としたタスクは、「UNIX-QA」に関するもの（UNIX コマンドのうち電子メール関係）で、語彙数は 521 単語である。構文知識を表わしている文脈自由文法は非終端記号 259，ワードクラス数（文法的に等価な終端記号の集合）268，書換規則数 534 からなっている。この他にワードクラスから終端記号への書換規則が 600 ある。こ

の文法によって生成される文数は約10^{37}個である。認識対象50文についての単語パープレキシティは10.7である。本タスクで認識する文例の一部を以下に示す。

(1) 現在作成したメールを山田さんへ送信してほしい。
(2) 鈴木氏にメールを送りたい。
(3) メールを暗号化して渡せるか。
(4) ファイルの内容を送信したい。
(5) メールが来ていることを知るにはどうすればよいか。

3.8.3 音声資料

音節単位のHMMを作成する際に、学習用として成人男性6名が発声した音韻バランスのとれた216単語とタスクに現われる外来語80語の296単語を用いた（計296語×6名＝1776語）。また、話者適応化のために各話者ごとに20文を用いた。これらのデータは、目視により音節単位にセグメンテーションされた。話者適応化では、出力確率分布の平均値だけを再学習している[22, 38]。また、単語発話速度に対する連続音声の発声速度の平均的な比（約0.8）で、HMMの継続時間長パラメータの分布を線形に縮めている（但し、One Pass法に基づくシステムSPOJUS-SYNO-Xでは元のままとした）。

認識用のデータは上記6名が発声したタスクに関する50文[9]であり（録音状態の悪い19発話文を除く計281文）、特に発声方法の指定はしていない。このため、文節間には休止区間はほとんど現われず（5文に一箇所程度）、一息に朗読されたものが多く、発声スピードは話者SNが9.0[モーラ/s]，話者TIが8.0[モーラ/s]，話者HUが7.7[モーラ/s]，話者KOが8.6[モーラ/s]，話者MAが8.9[モーラ/s]，話者SEが6.7[モーラ/s]である[10]。

3.8.4 SPOJUS-SYNO III（ワードスポッティング法ベース）の評価

(a) 実験条件

本章の実験では、特に記さない限りワードスポッティング法ベースのシステムに関しては次のような設定を用いた。

[9]平均文長は7.76単語。付録A.1参照。
[10]モーラは、韻律論において強勢や抑揚などの単位となる音の相対的な長さ。日本語では、ほぼかな1文字（拗音では2字）が1モーラに相当し、促音や撥音も1モーラである。

- ビームサーチ幅 = 45
- 枝刈りのしきい値$\lambda(i)$ = 300 （3.6.1節参照）
- 2単語間の接続時の許容範囲
  - 従来法（ラティスパージング手法）の場合： gap = overlap = 40フレーム
  - 連続DP法、O(n)DP法ベースの手法の場合： gap = overlap = 20フレーム
  - Bundleサーチ型手法の場合： gap = 5フレーム、overlap = 0フレーム
- ギャップ・オーバーラップ区間のペナルティスコア
  - 従来法（ラティスパージング手法）の場合： penalty = 250／フレーム
  - 連続DP法、O(n)DP法ベースの手法の場合： penalty = 400／フレーム
  - Bundleサーチ型手法の場合： penalty = 400／フレーム

ワードスポッティング法としてO(n)DP法を用いる場合には、前述した理由から最適単語系列に対するペナルティ値として、$C_p = -2$（1フレーム当りの対数値）を用いた。このペナルティ値は、実験によって文音声に対する最適単語系列（構文の制約なし）の対数尤度と正しい単語系列の対数尤度との差の値の平均的な値で適当に決定した。このペナルティを与えなかった場合は連続DP法によるワードスポッティング法よりも悪い結果となった。

従来のシステムでは、同じ話者適応化したHMMを用いて作成されたワードラティスを使って文認識を行った。二単語間の接続時の許容範囲は、単語の検出誤り等に対処するために大きめにとっている。

(b) 実験結果

文認識による従来のシステムとの認識率の比較を表3.2に示す。従来のシステムの結果のみ第2位までの認識率を括弧付きで示した。ワードスポッティング法として連続DP法を用いた場合は、従来のワードラティスの構文解析法による方法に対してワードラティスを用いない点を除いてほぼ等価であるが、単語候補の検出漏れが減るために文認識率が向上している。しかし、従来のシステムの第2位までの結果より悪いのは、ワードスポッティング法によるセグメンテーションの精度の影響が大きいことを示唆している。その結果を裏付けるように、ワードスポッティング時に言語制約（音節連鎖）を与えたことでセグメンテーション精度の向上が期待できるO(n)DP法に基づく方法では、連続DP法よりもよい結果が得られている。3番目の文レベル処理の結

表 3.2: ワードスポッティング法に基づく方法による文認識率 (%)
(括弧内は第 2 位までの文認識率)

話者	SN	TI	HU	KO	MA	SE	平均
スポッティング	64.6	89.4	76.1	78.3	85.7	86.7	80.1
＋構文解析法	(70.8)	(91.5)	(87.0)	(82.6)	(98.0)	(86.7)	(86.1)
連続 DP 法	75.0	87.2	87.0	78.3	87.8	84.4	83.3
O(n)DP 法	85.4	87.2	84.8	93.5	83.7	84.4	86.5
Bundle 法	93.8	89.4	87.0	91.3	89.8	91.1	90.4

果を用いたワードスポッティング法は、文認識率が 90%を唯一越え、最も良い結果が得られた。ギャップとオーバラップ共に 10〜20 とした時の 6 名平均の文認識率は 89.7 %で、ギャップとオーバラップを許さない場合は 87.2 %であった。これらのことから、より良いセグメンテーション精度を得るために言語情報の利用が重要であることが分かる。

3.8.5 SPOJUS-SYNO X（One Pass 法）の評価[11]

(a) 実験条件

本章の実験では、特に記さない限り One Pass 法ベースのシステムに関しては次のような設定を用いた。

- ビームサーチ幅 = 45（従来法も同じ）
- 予測制限及び枝刈りのしきい値$\lambda(i) = 250$

このシステムを用いた実験では、計算量の削減のために HMM の継続時間長制御を行わない場合も試みた。その場合は HMM の各状態に自己ループを設定し、全ての HMM について状態遷移確率 a_{ij} を次のように一定値とした。

$$a_{ij} = \begin{cases} 0.9 & (i = j\text{のとき}) \\ 0.1 & (i \neq j\text{のとき}) \end{cases} \tag{3.26}$$

(b) 実験結果

従来のワードスポッティングに基づく方法との比較を表 3.3に示す。6 名の話者の平均文認識率において、継続時間長制御を用いた One Pass 法の場合 90 %が得られており、従来の方法にくらべて 9.9 %向上している。表の第 2 位の認識結果は、One Pass 法の認識処理の中で、各オートマトンの状態毎に 2 番目に高いスコアとなる単語候補のスコアを記憶しておくことによって、近似的に求めている。この結果、半分近く文認識誤りが減少しているが、それは多くの誤りが短い助詞などの単語の誤りであったためである。

[11] 4章以降で扱う「富士山観光案内」タスクにおける認識実験結果を付録 B.6に示した。

表 3.3: 従来の方法と One Pass 法の文認識率 (%)（括弧内は第 2 位まで）

話者	SN	TI	HU	KO	MA	SE	平均
スポッティング	64.6	89.4	76.1	78.3	85.7	86.7	80.1
＋構文解析法	(70.8)	(91.5)	(87.0)	(82.6)	(98.0)	(86.7)	(86.1)
One Pass 法	91.7	91.5	89.1	91.3	87.8	88.9	90.0
	(97.9)	(97.9)	(91.3)	(95.7)	(89.8)	(93.3)	(94.3)
One Pass 法	81.3	91.5	89.1	89.1	83.7	91.1	87.5
(継続時間長制御無し)							

(c) ビームサーチ法及び予測制限等による高速化の効果

予測制限及び枝刈りのためのしきい値は、3.6.2節で説明したように、図 3.8で示される正解文の最適照合パスの単語境界の累積対数尤度とその時点の累積対数尤度の最大値との差の分布に従って、この分布をカバーするように次のように設定した。

$$\lambda(i) = 250.0, 200.0, 150.0, 100.0 \quad (200\text{ フレーム毎に変更、最小値} = 100.0) \tag{3.27}$$

表 3.4に、構文解析による予測のためのビームサーチの制限と、One Pass 法での枝刈りによる計算量削減の効果を示す。文認識率を低下させない範囲で評価を行ったため、文認識率は表 3.3 の結果と同じである。1 フレーム当りの照合単語数は計算量を知るための目安となり、生成された有限状態オートマトンの状態数は、必要な記憶容量に関係する。表の「ビームサーチ幅可変」は、予測のためのビームサーチで、枝刈りの場合と同じように累積対数尤度しきい値を用いた予測制限を行った場合である。この場合、1 文全体で生成された状態数は、3／4 程度に減少している。また、1 フレーム当りの単語照合数は、Viterbi サーチでの枝刈りも併せて行うことで約 1／8 に減少しており、タスクの語彙数 521 よりも少なくなっている（すなわち、オリジナルな Viterbi アルゴリズムによる拡張連続 DP 法よりも高速になる）。枝刈りのしきい値は、一定値と

表 3.4: ビームサーチ等による計算量の削減効果
（HMM 継続時間長制御なし。6 名話者の平均。）

ビームサーチ幅及び照合時の枝刈り	#WDVR.AV.	#GEN.ST.	認識時間 [s]
固定 (45)，枝刈無し	2,458	863	363
可変 ($\leq$ 45)，枝刈無し	2,130	611	316
可変 ($\leq$ 45)，枝刈有り	319	606	121

使用計算機　:OMRON LUNA-88k (25MIPS)
#WDVR.AV.:一フレーム当りの照合単語数（一文平均）
#GEN.ST.　:生成された状態数（一文平均）

したときと、階段状の単調減少形に設定したときで、ほとんど差は無かった。これは、処理フレームが進むと構文的な制約から不適当な部分文が生成されにくくなり、枝刈りの効果が目立たなくなるためと考えられる。

(d) N-Best の文認識の検討

本来、One Pass 法においてスコアが良い順に上位 N 個の解を得るためには、約 N 倍の記憶および認識時の計算が必要となる。N-best を効率的に求める方法の一つとして、Soong らは Tree-trellis N-best 探索法を提案している[16]。Soong らの方法は、前向き（フレーム同期）の探索と後向き（フレーム非同期）の探索を組み合わせ、前向きのスコアを用いた後向きの A*探索により N-best 解を求める方法であり、必要なだけの N-best の解を順次求めることができるという特徴がある。ここでは、本来の 1-Best の計算に対してわずかな処理を加えて近似的に N-Best を求める方法を試みた。この方法は、Schwartz らにより Lattice N-Best アルゴリズムとして提案されている方法と同じである[19]。3.5.2節のアルゴリズムで、オートマトンの状態毎に、その状態に遷移する全単語の累積スコアとバックトレースの情報を記憶しておくすなわち、従来のアルゴリズムで最もスコアの良い単語でのみ残していた $P_q(i)$, $B_q(i)$, $N_q(i)$ を、全単語について記憶する。そして、文末でのバックトレース処理において、オートマトンの状態に記憶されている全てのバックトレース情報を用いて N-Best の文候補を求める。このとき、得られる文候補の数は莫大になるので、最良の文候補のスコアとの差や各状態のバックトレースの分岐数を制限する。この方法で複数の候補が得られる過程を、概念的に図 3.14に示す[19]。

本実験では、$P_q(i)$, $B_q(i)$, $N_q(i)$ に加えて 2 番目に累積スコアが良い単語についての情報（$P_q^{2nd}(i)$, $B_q^{2nd}(i)$, $N_q^{2nd}(i)$）だけを記憶し、上述の手順で第 N 位の文候補を求めることを検討した。このバックトレース処理では、オートマトンの状態遷移系列が第一位のパスと異なる場合も含めて探索を行なう。このような方法で第 5 位までの認識を行なった時の文認識率を表 3.5に示す。なお、この実験では HMM の離散継続時間分布を 3 乗して重み付けをしたことにより、1 位の結果でもこれまで示した結果より良い認識率が得られている。

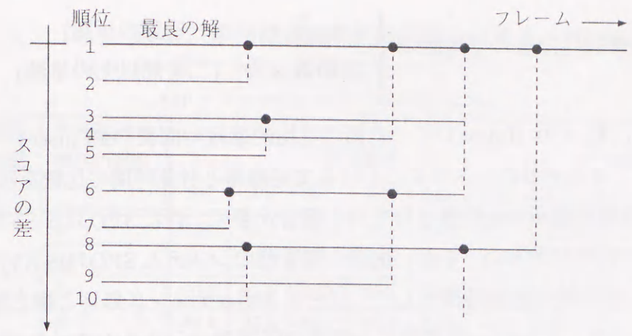


図 3.14: N-Best 候補のバックトレース処理[19]
（左側の 1~10 に示されている列がそれぞれ順位毎の文候補に対応する。ドットの部分は、バックトレース時にあるオートマトンの状態で 1-Best 以外の単語 を選択した場合の分岐を表す。）

表 3.5: N-Best 文認識の結果
（HMM 継続時間分布を 3 乗して重み付け）

順位	SN	TI	HU	KO	MA	SE	平均	誤認識文総数
1 位のみ	95.8	93.6	89.1	91.3	91.8	95.6	92.9	20
1~2 位	97.9	97.9	93.5	100.0	98.0	95.6	96.8	9
1~3 位	97.9	97.9	97.8	100.0	98.0	97.8	97.9	6
1~4 位	97.9	97.9	97.8	100.0	98.0	100.0	98.2	5
1~5 位	97.9	97.9	97.8	100.0	98.0	100.0	98.2	5

3.8.6　単語継続時間長制限の効果

One Pass 法ベースのシステム SPOJUS-SYNO-X と、ワードスポッティング法ベースの Bundle 型サーチ法のシステム SPOJUS-SYNO-III の両方について、単語継続時間長の下限値の制限を以下のように設定してその効果を調べた。

$$\text{単語継続時間長の下限値} = \begin{cases} \frac{T_0}{3} & (\text{一音節の単語}) \\ \text{音節数} \times \frac{T_0}{2} & (\text{二音節以上の単語}) \end{cases} \tag{3.28}$$

但し、$T_0 = 24$ (frame) = 一音節の平均的継続時間長 (120 msec)

表3.6に、それぞれのシステムにおける文認識率と計算時間の比較を示す。両者とも、認識率と計算時間が少し改善されている場合が多い。特にSPOJUS-SYNO·Xでは、計算時間での効果が大きい。なお、従来の階層型のシステムSPOJUS-SYNO·IIでは、一文あたりの平均的な計算時間として、ワードスポッティング処理と構文解析処理がそれぞれ約10分要していた。結果的に、従来の階層型のシステムに対して共に1/5~1/10の計算時間となっている。このような改善のおもな要因であるビームサーチ法や枝刈りの効果と比べると、単語継続時間長の制限による計算量削減の効果はそれほど大きくはないが、先見的な知識として容易に適用できる点で有用である。

表3.7には、One Pass法に基づくシステムにおける計算量の比較を示す。前に述べた枝刈り等を用いた上で評価しているが、単語継続時間長の制限によって、照合単語数、生成された状態数とも、更に、それぞれ2割と1割程度減少している。継続時間長制御付きの音節HMMを使用した場合にも同様に効果が得られ、文認識率も若干（1.8%）向上した。結果的には、構文解析による状態の予測の制限として有効であることが示され、比較的容易に計算時間の短縮の効果が得られた。また、単語継続時間長制限と直接関係はないが、HMMの離散継続時間分布の確率を3乗して重みづけを行ったところ（音節継続時間の制約の強化）、平均文認識率は92.5 %となり、若干の計算量の削減の効果も見られた。これも、音節単位の継続時間の制約によって、単語単位の継続時間の制約が働いたためと考えられる。

3.8.7　オートマトン展開における効率化の評価

3.7.2節で述べた、文法のあいまいさに伴う単語照合の重複に対処するためのオートマトンの状態生成の改良の効果について調べた。ここでの実験は、特に計算量の面で効果が期待できるOne Pass法に基づくシステムにおいてのみ行なった。表3.8は、6名のテストデータに対する文認識実験の結果から、一文当りの計算量の比較を行なったものである。従来法とは、一つの文法上のパスを一つのオートマトンの状態に対応付ける方法で、改良法とは、前述の方法で文法上のパスのある集合毎に一つのオートマトンの状態に対応付ける方法をとった場合を示している。

この実験結果では、計算量および記憶量が約半分位にまで改善されている。但し、改良された方法では単語列の重複した照合をしないので、ビームサーチ幅を半分以下に絞っている。後に表3.12に示すように、改良された方法ではビームサーチ幅を10程度

表 3.6: 単語継続時間長の制限の効果

(a) Bundle 型サーチ法（ワードスポッティング法ベース）
(gap = 5, overlap = 5, $\lambda(i)$ =式 (3.27))

HMM 継続時間長制御	単語継続時間長の制限	文認識率（%）							計算時間 [s]
		SN	TI	HU	KO	MA	SE	Ave.	
なし	なし	77.1	85.1	84.8	84.8	79.6	91.1	83.6	71
	あり	81.3	87.2	87.0	84.8	79.6	91.1	85.1	63
あり†	なし	89.6	91.5	89.1	89.1	89.8	91.1	90.0	97
	あり	89.6	91.5	89.1	89.1	91.8	93.3	90.7	98

使用計算機:　OMRON LUNA-88k (25MIPS)
†発声速度を考慮した継続時間分布の線形な伸縮を行なっていないため、表3.2の実験と条件が少し異なる。

(b) One Pass 型サーチ法 ($\lambda(i)$=式 (3.27))

HMM 継続時間長制御	単語継続時間長の制限	文認識率（%）							計算時間 [s]
		SN	TI	HU	KO	MA	SE	Ave.	
なし	なし	81.3	91.5	89.1	89.1	83.7	91.1	87.5	121
	あり	83.3	89.4	89.1	89.1	83.7	91.1	87.5	102
あり	なし	89.6	91.5	89.1	89.1	89.8	91.1	90.0	305
	あり	91.7	91.5	91.3	91.3	89.8	95.6	91.8	249
あり 重みつき†	あり	95.8	91.5	89.1	91.3	91.8	95.6	92.5	227

使用計算機:　OMRON LUNA-88k (25MIPS)
†重みつき: HMM 継続時間長分布を 3 乗

表 3.7: 単語継続時間長の制限による計算量の比較
(One Pass 型サーチ法, 6 名話者の平均, $\lambda(i)$=式 (3.27))

HMM 継続時間長制御	単語継続時間長の制限	#WDVR.AV.	#GEN.ST.	認識時間 [s]	文認識率 (%)
なし	なし	319	606	121	87.5
	あり	260	570	102	87.5
あり	なし	288	608	305	90.0
	あり	226	566	249	91.8
あり 重みつき†	あり	209	534	227	92.5

使用計算機　:OMRON LUNA-88k (25MIPS)
#WDVR.AV.:一フレーム当りの照合単語数（一文平均）
#GEN.ST.　:生成された状態数（一文平均）
†重みつき　:HMM 継続時間長分布を 3 乗

表 3.8: オートマトン展開法の改善による単語照合重複への対処の効果
（HMM 継続時間制御なし。$\lambda(i)$=式 (3.27)。）
最大ビームサーチ幅：45 (従来法), 20 (改良法)

	S.Acc.(Ave.)[%]	#WDVR.AV.	#GEN.ST.	計算時間 [s]
従来法（NFA 生成）	87.5	319	606	121
改良法（DFA 生成）	87.5	115	272	81

（使用計算機　:OMRON LUNA-88k（25MIPS）
#WDVR.AV.:一フレーム当りの照合単語数（一文平均）
#GEN.ST.　:生成された状態数（一文平均））

まで絞っても認識率がほとんど変わらないため、従来に比べてかなり効率化が図れることが分かる。なお、ワードスポッティング法ベースのシステムでは実験を行なっていないが、アルゴリズムの特徴から単語照合レベルの計算量にはあまり影響しないため、この実験結果に比べると効果が小さいと考えられる。しかし、文法のあいまいさによって必要となるビームサーチ幅は数倍に増大するので、現在でも比較的処理量が多いところの、部分文仮説を生成する際のソーティングや、部分文仮説と単語の接続のチェックなどに要する処理の効率に対する影響は大きいといえる。したがって、特にタスクの規模が大きくなるほど、ワードスポッティング法ベースのシステムでも同様な改良が有効になると考えられる。

3.8.8　SPOJUS-SYNO の性能の比較

従来のシステムと今回評価を行なった二つのシステムについての実験結果を表 3.9に示す（表 3.2、表 3.3 参照）。ワードスポッティングに基づく方法 (SPOJUS-SYNO III) に関しては、前述の実験で最も良い認識精度が得られている Bundle サーチ型手法の結果だけを示している。

HMM の継続時間長制御を行なった場合、提案した手法は共に約 90 %の文認識率が得られ、従来の方法に対して文認識誤りが半分近くに減少していることが分かる。

HMM の継続時間長制御を行なわない場合は、今回の二つの方法についてのみ評価を行なっている。結果を表 3.10に示す。この条件では、2 つのアルゴリズムの性能の差が現れ、Bundle サーチ型手法よりも One Pass サーチ手法の方が認識精度が良いという結果になった。これは、単語の継続時間長の制限が緩くなったことで、前者のアルゴリズムが単語境界の最適な点を近似して求めていることの影響が出たためと考えられる。

表 3.9: システムの文認識率の比較
（HMM 継続時間長制御あり）

話者	SN	TI	HU	KO	MA	SE	平均
スポッティング	64.6	89.4	76.1	78.3	85.7	86.7	80.1
+構文解析法	(70.8)	(91.5)	(87.0)	(82.6)	(98.0)	(86.7)	(86.1)
Bundle 型サーチ	93.8	89.4	87.0	91.3	89.8	91.1	90.4
One Pass サーチ	91.7	91.5	89.1	91.3	87.8	88.9	90.0

表 3.10: システムの文認識率の比較
（HMM 継続時間長制御なし。gap = 5, overlap = 5 (Bundle 型)）

話者	SN	TI	HU	KO	MA	SE	平均
Bundle 型サーチ	77.1	85.1	84.8	84.8	79.6	91.1	83.6
One Pass サーチ	81.3	91.5	89.1	89.1	83.7	91.1	87.5

次に、文認識率が良い One Pass アルゴリズムと Bundle 型ワードスポッティング法を用いたアルゴリズムの計算量の比較について述べる。図 3.15は、それぞれのアルゴリズムで、実験に用いた各テスト文において実際に Viterbi 法による照合が行われた単語数を 1 フレーム当りの平均でプロットしたものである。構文制御による One Pass アルゴリズムでは最適な照合を行うため、同一単語に対して複数の照合が行われることがある。従って平均的な照合単語数は語彙数を越える場合もあり、発話内容による変動も大きいことが分かる。一方、後者ではワードスポッティング法に基づいており、文レベル処理の結果予測された単語だけしか処理を行わないため、照合単語数は常に語彙数 (521 単語) よりも少ない上に各文毎の変動も小さい。スポッティング処理の中断を含めた単語候補の制限を行った場合、さらに 3 分の 1 程度に減少しているのが分かる。

この結果から、ワードスポッティング法を用いた場合、音声照合のための計算量に関してはかなり少なくて済むといえる。しかし、3.6.1 節で述べたように単語列の生成や文仮説スコアの計算なども必要であり、単語候補がかなり減ると影響は小さいが、若干は認識時間に関係してくる。実験で認識に要した時間 (OMRON-LUNA88k,CPU-time) は、One Pass アルゴリズムの場合は 6 名平均で 1 文当り約 5 分、ワードスポッティング法を用いた場合は約 1.5 分であった（初期化等の処理を含む）。HMM の継続時間長制御を省略すれば、それぞれ 3 倍と 2 倍近く高速化できた。

3.8.9　探索処理単位としての文法記述単位の比較・評価

これまで用いていた文脈自由文法で記述された言語モデルは、終端記号の単位を単語としている。ここでは、単語単位で記述された「電子メール」タスクの文法を音節単

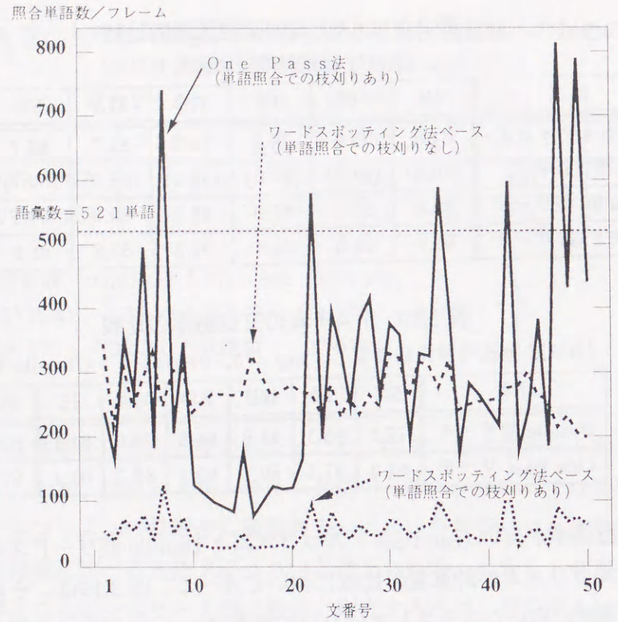


図 3.15: 単語レベルにおける音声照合の計算量の比較（話者: HU）

位の文法へ変換し、認識処理の計算量などを比較するための文認識実験の結果を示す。

音節単位で記述される文法への変換は、以下のような手順で行なった。まず、従来のワードクラスから単語への書き換え規則を、音節の列への書き換え規則に変換し、終端記号と非終端記号だけからなる文法を作成した。そして、音声認識処理の計算量を軽減するために、ワードクラス内に限って音節系列の規則を最適化した。最適化はワードクラス毎に、一旦有限状態オートマトンの形にして、それを決定性オートマトンへ変換し、決定性オートマトンの最簡形を求めた後[40]、元の書き換え規則に戻すことにより実現した。一例として、最適化の前後のワードクラスから音節系列への書き換え規則を図 3.16に示す。

元の単語単位で記述された文法と、上述の手順で音節単位へ変換された文法の規則数等を比較した結果を表 3.11に示す。音声単位で記述された文法では、終端記号として文法の語彙（521 単語）に使われる音節だけを登録しているため、全音節カテゴリー数の 110 個より少なくなっている。これらの２種類の文法を使用し、認識率とその計算効率を調べた結果を表 3.12及び図 3.17に示す。図 3.17は、表 3.4での計算量の比較のときと同様に、一フレーム当りの照合単語／音節数 (#WDVR.AV.) と、一文当りの生成された状態数 (#GEN.ST.) を示している。 なお、ここでは 3.7.2節で述べたオートマトン展開法の改良を行なったシステムを用いており、ビームサーチ（単語レベルの照合）の効率化が為されている。表 3.12で、例えば平均認識が両者で同じく 92.9%となっている点を比べると、それぞれビーム幅が 10（単語単位）と 80（音節単位）であり、音節単位の方がかなりビームサーチ幅が多く必要となっている。一方、単語又は音節レベルでの計算量を比較した図 3.17を見ると、照合単語（音節）数はほぼ同じであることから、音節単位の方が計算量が少ないことが分かる（テストデータにある単語は一単語平均約 2.5 音節）。しかし、結果的に計算時間の差がそれほど大きくなかったのは、音節単位の文法を用いた場合はオートマトンの状態数が 3 倍弱生成されるために状態の生成に伴う処理と予測回数が増加したことと、ビームサーチ幅を大きくとることからソーティング処理の増加が無視できなくなったためである。

現在扱っている全音節数が 110 個であることを考えると、音節単位の文法の場合にビームサーチ幅が 80 程度というのはかなり大きいが、実際は累積尤度スコアによる制

file → bu N ke N （文献）
file → bu N me N （文面）
file → bu N syo （文書）
file → e e bu N （英文）
file → fa i ru （ファイル）
file → wa bu N （和文）

⟹

@FIL1 → bu N @FIL11
@FIL1 → e e @FIL12
@FIL1 → fa i ru
@FIL1 → wa @FIL12
@FIL12 → bu @FIL13
@FIL11 → ke @FIL13
@FIL11 → me @FIL13
@FIL11 → syo
@FIL13 → N

図 3.16: ワードクラス *file*の書き換え規則の音節単位の最適化手順
（左が最適化前、右が最適化後）

表 3.11: 文法の規則数等の比較

文法の記述単位		単語単位	音節単位
終端記号数		521	83
非終端記号数		259	1292
ワードクラス数		268	0
書換え規則数	主要部	534	1292
	ワードクラス	600	0
テストセット perplexity（単語／音節）		10.7	2.4

表 3.12: 文法記述単位の違いによる文認識率・処理時間の比較

- HMM 継続時間制御あり（３乗して重み付け）
- 括弧内の数字は認識結果候補が１つも得られなかった入力文数

(1-a) 単語単位で記述された文法の場合（しきい値$\lambda(i)$ =−250）

ビーム幅	SN	TI	HU	KO	MA	SE	ALL	TIME[s]
3	87.5(1)	93.6	89.1	91.3(1)	87.8	95.6	90.8	105
5	93.8	93.6	89.1	91.3	87.8	95.6	91.8	123
10	95.8	93.6	89.1	91.3	91.8	95.6	92.9	135
20	97.9	93.6	89.1	91.3	91.8	95.6	93.2	143

(1-b) 音節単位で記述された文法の場合（しきい値$\lambda(i)$ =−250）

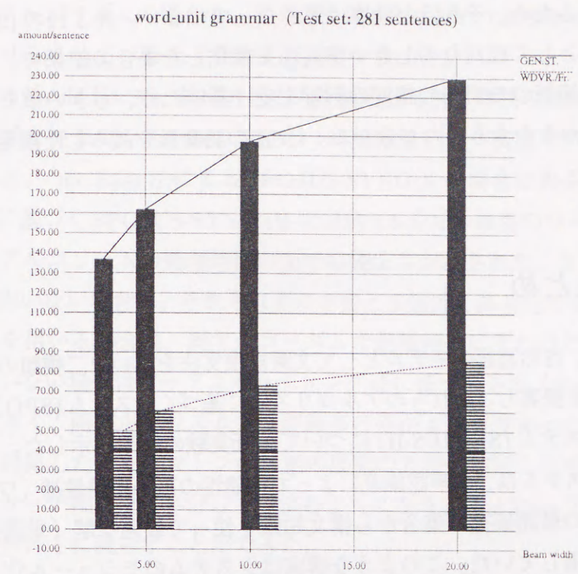
ビーム幅	SN	TI	HU	KO	MA	SE	ALL	TIME[s]
10	87.5(2)	89.4	84.8(1)	89.1(2)	87.8	95.6	89.0	92
20	91.7	89.4(1)	89.1	91.3(1)	91.8	95.6	91.5	105
40	93.8	91.5	89.1	91.3	91.8	95.6	92.2	113
60	93.8	93.6	89.1	91.3	91.8	95.6	92.5	121
80	95.8	93.6	89.1	91.3	91.8	95.6	92.9	121

(2-a) 単語単位で記述された文法の場合（最大ビーム幅= 10）

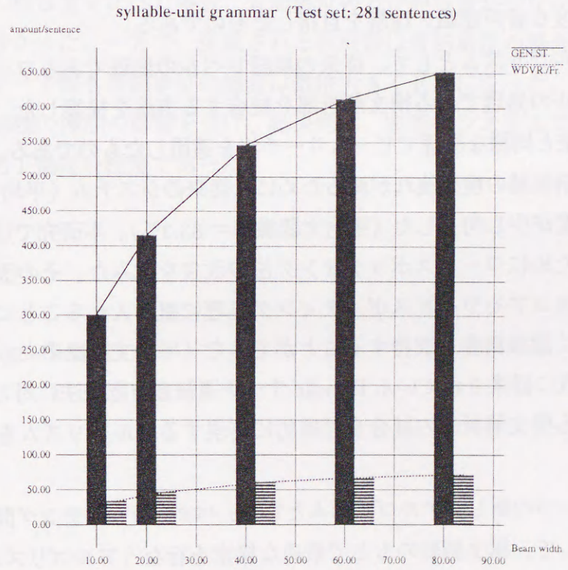
しきい値$\lambda(i)$	SN	TI	HU	KO	MA	SE	ALL	TIME[s]
-100	79.2(7)	89.4(3)	82.6(2)	84.8(2)	73.5(6)	93.3	83.6	84
-150	95.8(1)	91.5	89.1	91.3(1)	83.7(2)	95.6	91.1	96
-250	95.8	93.6	89.1	91.3	91.8	95.6	92.9	135

(2-b) 音節単位で記述された文法の場合（最大ビーム幅= 80）

しきい値$\lambda(i)$	SN	TI	HU	KO	MA	SE	ALL	TIME[s]
-100	72.9(6)	87.2(4)	80.4(2)	84.8(2)	71.4(8)	91.1(1)	81.1	79
-150	93.8(1)	89.4(2)	89.1	91.3(1)	83.7(3)	95.6	90.4	90
-250	95.8	93.6	89.1	91.3	91.8	95.6	92.9	121



(a) 単語単位の文法における計算量



(b) 音節単位の文法における計算量

図 3.17: 文法の記述単位（単語及び音節）による計算量の比較

限（枝刈り）によって一フレーム平均のビームサーチ幅の使用率としてはおよそ1／3以下であるため、それほど効率は悪くなっていない。表3.12の(2-a),(2-b)は、ビーム幅を固定として枝刈りのしきい値$\lambda(i)$を変化した場合の結果で、音節単位の文法の場合は、音節毎のモデルの精度の影響を受け易いため、しきい値を厳しくした時の認識率の低下がやや大きいのが分かる。(文法の語彙数を減らした補足実験の結果を付録A.2に示した。)

3.9 まとめ

本章では、自然言語のモデルとして文脈自由文法を用いた二種類の連続音声認識のアルゴリズムを提案し、それらのアルゴリズムに基づくシステム(SPOJUS-SYNO·III/X)と従来のシステム(SPOJUS·II)について評価実験の結果を示した。

従来のシステムは、音声認識部によって可能性の高い単語候補（ワードラティス）を出力し、その単語候補の集合から構文知識を使って最適な解（単語列）の探索を行なう方法を採用していた。このような構成はシステムのモジュール化という点で有効であるが、中間的な情報の損失が起こり得ることが予想された。ここで提案した手法は、単語等の音声認識のより低レベルの処理においても構文知識を利用することによって、さらに高精度な音声認識／理解を目指したものである。

新しいアルゴリズムとして、従来の単語レベルの処理であるワードスポッティング部と文レベルの処理である構文解析部を統合する方法を提案した。その実現方法は拡張連続DP法と同様な原理でビームサーチ法を適用したものである。この方法によって中間的な単語候補の検出洩れが減るために、従来のシステム（平均文認識率＝80.1%）に比べて精度が少し向上した（平均文認識率＝83.3%）。本研究では、更に文認識精度を改善するためにワードスポッティング法の改良を試みた。その改良のなかで、文認識レベルのスコアをワードスポッティング処理に組み入れることによって、これまでよりも大きく認識精度を改善することができた（平均文認識率=90.4%）。この方法は結果的に、既に提案されているBundleサーチ連続音声認識法に対して、新たに文脈自由文法による構文解析部の統合を効率的に実現するアルゴリズムを提案したものといえる。

更にもう一つの新しいアルゴリズムとして、パターンマッチング問題としての連続音声認識に対して、構文制約のもとで最適な探索を行なうアルゴリズムであるOne Pass DP法を適用する方法を提案した。この場合、構文制約としては有限状態オートマトンを用いなければならないが、本論文では従来のトップダウン型構文解析法を用いてオートマトンの状態を動的に展開するための手順を加えることにより、One Pass DP法において文脈自由文法による言語制約を効率的に適用できる準最適なアルゴリズムを示した。このアルゴリズムの場合も、従来のシステムに比べて結果的に10％近く文認識率を向上させることができた（平均文認識率＝90.0%）。

両方のアルゴリズムにおいて、枝刈り法などを適用して計算時間の短縮を検討し、平均的な文認識時間を従来のシステムの数分の一に短縮した。一文当りの平均的な認識時間で比較すると、One Pass法によるSPOJUS-SYNO·Xの場合は約5分、ワードスポッティング法に基づくSPOJUS-SYNO·IIIでは約1.5分で、後者のワードスポッティング法に基づくアルゴリズムの処理時間の面での優位さが示された。またHMMの継続時間長制御を用いない場合は、それぞれ更に3倍と2倍近く高速化できた。HMMの継続時間長制御を用いる場合は、両アルゴリズムで認識精度にそれほど差がない結果であったが、用いない場合はワードスポッティング法に基づく方法（Bundle型）は認識率の低下が大きく、両アルゴリズムで文認識率の差が4％程度あった。また、文法のあいまいさに対処したオートマトンの状態の展開の方法を検討したが、この方法では必要なビームサーチ幅が更に半分以下で済むようになり、One Passアルゴリズムでは計算量を約3分の1に減らすことができた。

認識アルゴリズムの検討とは別に、文法記述の単位を音節又は単語とした場合の認識効率の違いの評価を行なった。文法記述単位に関して文認識実験によって調べた結果は、終端記号の照合量を比べた場合、音節単位の方が照合量が少ない（およそ半分くらい）ことが分かった。一方で、生成される状態数は音節単位の場合の方が倍以上になり、ビームサーチ幅も音節単位の場合は数倍大きく必要になった。そのため、それらに関係した処理の増加の影響を受け、結果的に、実際のシステムでの認識時間は音節単位の方が約1割短い程度であった。

第 4 章

自然な発話における未知語・不要語の処理

4.1　はじめに

これまでの音声認識システムは、一般に読み上げ音声（朗読音声）を対象としてきた。しかし、対話音声の認識・理解のためには、自由な発話に特有な音声現象を扱えるようにしなければならない。特に、現在のシステムで問題となるのは、間投詞、言い直し、言い淀み、未知語、倒置、文法を逸脱した発話などの現象である。ここでは、その中で間投詞や未知語などを処理するためのアプローチについて考える。

未知語や冗長な語を含む発話を扱うための方法としては、これまでに garbage モデルを使用する方法[55]や、音韻連鎖モデルによる方法などが報告されている[56, 63, 64, 58]。garbage モデルは、制約のない発話からキーワード抽出を行なうようなタスクに多く適用され[55, 58]、最近では連続音声認識での適用も報告されている[65]。しかし、様々な間投詞の単語や非音声などの音響的な特徴を少数のモデルで表現するため、セグメンテーションが困難な連続音声認識や、特に大語彙で発話の制約が少ないタスクを扱う場合に十分な未知語の検出性能が期待できない[58]。また音声対話システムにおいては、確認 (verification) の対話やリジェクションなどのために、より正確な未知語の検出法が望まれる。本研究では、基本的に後者のアプローチに基づき、音節単位の音響的モデルによって未知語検出を行なう方法について検討している。また、後述のように間投詞の単語を未知語として扱うことで、効率的に処理する方法についても検討している。ところで、同様な方法は、文法外の発話を棄却する方法として有効であることが示されている[66]。ここで述べる未知語処理法も、文法外の発話や孤立単語認識における未知語の棄却の方法として適用できるが、そのような観点からの評価について

は5章で述べる。また、ここで述べる未知語処理法による未知語の棄却性能と認識性能との関係についても5章で触れる。

間投詞は音声認識システムにおいては一般に冗長語（不要語）とみなされ、意味理解に用いられることはほとんどない[1]。しかし、実際に対話で観測される間投詞の単語の種類はかなり多いうえに発音があいまいな場合が多いので[69, 70]、音声認識ではそのような問題に対処できるようなアプローチを考える必要がある。ここでは、多様な間投詞の単語のほとんどを未知語として処理する冗長語処理を検討する。後述する未知語処理法の一つは、構文的に多く“未知語”仮説が予測されるような場合にも処理量の増加が非常に少ない方法であり、処理量の点で特に有効な手段といえる。

本章では、上述のように、未知語や間投詞の扱いに対して未知語処理法を用いたアプローチを検討する。始めに、音声認識システムの音声・言語処理での冗長語の扱いについて考えるために、実際の音声対話における間投詞の特徴分析の結果について述べる。次に、厳密な未知語・冗長語処理のアルゴリズムとその近似的な実現法について述べる。最後に、未知語・冗長語処理を組み込んだ音声認識システム SPOJUS-SYNO·Y による、未知語・間投詞を含んだ発話の評価実験の結果を示す。

4.2 音声対話における間投詞の扱い

ここでは、特に音声認識システムでの間投詞の扱いを考えるために、音声対話における間投詞の出現傾向の調査結果について述べる。なお、この特徴分析の結果は文献[69]に基づいている。間投詞以外の自然発話特有のおもな現象であるポーズや言い直しに対する分析結果についてはそちらを 参照されたい。

調査のための対話データは、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の一部を使用している。調査対象の対話は、TUT001、TUT002、TUT003、TUT004、TUT005、TUT006、TUT008、TUT009、TUT0010、TUT0011 の計10対話である。話者数などのデータを表4.1に示す。

分析されたデータ中の、助詞落ち、倒置、間投詞、言い直しの出現頻度を表4.2に示す。この表から、間投詞はその出現頻度が顕著に多く、音声対話では特に現れやすい音声現象であることが分かる。この分析結果は人間どうしの対話を調査したものであるが、最近では、実際の音声対話システムを用いたり、ユーザに分からないように人間がシステムの応答を肩代りして対話を行なう Wizard of Oz(WOZ) 法を用いて、対話データの収集・分析が行なわれている[71, 72]。黒岩ら[71]の内線電話受け付けシステム

[1]間投詞や韻律的な情報の役割についての検討は少ないが、自然発話の頑健な認識・理解手法を実現するうえで重要になるものと思われる[73, 74]。

表 4.1: 分析データベース

ドメイン	観光案内, 相談, 他の案内
対話数	10 対話
話者数	11 人
総文数	1,052 文
文節数	4,597 文節

表 4.2: Ill-formedness の出現数

Ill-formedness	出現個数
助詞落ち	99 個 (0.094/1 文)
倒置	18 個 (0.018/1 文)
間投詞	1,185 個 (1.126/1 文)
言い直し	153 個 (0.145/1 文)

による分析結果では、1 文あたりの間投詞数が 0.073 個、上條ら[72]の WOZ 方式による道案内に関するシステムでは 1 文あたり 0.37 個（間投詞が含まれる発話の割合は全体の 26%）となっている。これらの結果は、人間どうしの対話の場合に比べてかなり間投詞の使用が少ないといえるが、その原因は対話の相手が機械になったという直接的な理由だけではなく、タスクやシステムの対話の実現の方法に依存していると考えられる。間投詞の文中の出現位置については、文頭に多くなり（人間どうしの場合は総間投詞数の約 40%[69]、機械との対話の場合は約 80%[71, 72]）、単独で発声される割合が多いという分析結果が得られている。また、システムの使用を継続するにつれて、間投詞の使用頻度が減少しているという事実も観測されている[71]。

表 4.1の対話データにおいて、間投詞が単独または幾つか連続して観測された場合の比率を表 4.3に示す。この結果から、間投詞は単独で発声される割合が多いことが分かる。

間投詞の種類数に関しては、表 4.1と同じ日本音響学会の模擬対話の書き起こしテキスト 5,558 文（87 対話）に対する調査で、424 種類の間投詞が使われていることが報告されている[65]。また別の対話データで、「国際会議の申込に関する参加者と事務局の対話」の模擬会話の 11,054 文（3,718 対話）に対する調査では[67]、およそ 112 種類（但し語尾の長音の違いも人の判断により区別）の間投詞が示されている。表 4.1の対話データで単独に現れた主な間投詞の内訳を表 4.4に示す。なお、間投詞に含まれる長母音や促音は無視して集計されている。上述のように間投詞の種類が多いのに対して、

表 4.3: 間投詞の連続発声数

連続数	出現数	[%]
1	1,044	88.1
2	117	9.9
3	22	1.9
4	2	0.2

表 4.4: 単独間投詞の内訳

間投詞	出現数	[%]	累積 [%]
え	238	22.8	22.8
えと	230	22.0	44.8
あの	207	19.8	64.7
あ	162	15.5	80.2
ま	130	12.4	92.6
えとですね	33	3.2	95.8
ん	9	0.9	96.6
そですね	9	0.9	97.5
その	7	0.7	98.2
こ	5	0.5	98.7
あと	3	0.3	98.9
んと	2	0.2	99.1
は	2	0.2	99.3
その他	7	0.7	100.0

一部の間投詞だけが多用されていることが分かる。上位5種類の間投詞で、単独で発声される場合の93%、間投詞の出現総数の82%を占めている。この上位5種類の間投詞は、長母音の違いを考慮した出現頻度による上位10種類にも対応している[70]。

4.3 サブワードモデルを用いた未知語・冗長語の処理

ここでは、Asadiらの未知語処理の方法[56]と同様に、サブワード単位の音響的モデル[2]を用いて未知語処理を実現する方法について述べる。

[2]サブワード (subword) 単位の音声のモデルとしては phonelike units(PLUs), syllable-like units, demisyllable-like units, acoustic units などがある[52]。

4.3.1 未知語処理の原理

本研究では文中の間投詞は発話の意味理解において重要ではないと考え、未知語として扱う。この場合、あらかじめ言語モデルにおいて間投詞の出現を仮定するだけでよい。前の節で示した間投詞の出現頻度の分析結果では、ごく少数の間投詞の単語が非常に多く現れることが示されているので、そのような単語だけ特別に辞書登録することも可能である。例えば、表4.4の上位5個だけでも使用される間投詞の単語をかなりカバーすることが期待できる。言い直しや言い淀みなどの不要語も同様に扱うことが考えられるが、それらを扱った場合の評価実験については6章で述べる。

音響・言語知識による音声の照合の過程は、次式で表される発話文の事後確率の最大化問題として考えることができる[60]。

$$P(\boldsymbol{W}|\boldsymbol{A}) = \frac{P(\boldsymbol{A}|\boldsymbol{W})P(\boldsymbol{W})}{P(\boldsymbol{A})} \tag{4.1}$$

但し、$\boldsymbol{A}$は音声の時系列入力、$\boldsymbol{W}$は認識結果として与えられる言語表現で、一般に単語列である。この式で $P(\boldsymbol{A}|\boldsymbol{W})$ は一般に HMM などによる音響的モデルに対応し、$P(\boldsymbol{W})$ は言語モデルに対応する。$P(\boldsymbol{W}|\boldsymbol{A})$ は、音響・言語モデルによる、$\boldsymbol{W}$の認識結果としての信頼性の尺度と考えることができるので、このような尺度に基づいて未知語の検出を考えるとよい[61, 62]。しかし、一般に $P(\boldsymbol{A})$ を直接推定することは難しいので、次のような近似を考える。

$$\hat{P}(\boldsymbol{W}|\boldsymbol{A}) = \frac{P(\boldsymbol{A}|\boldsymbol{W})P(\boldsymbol{W})}{\sum_{\boldsymbol{p}} P(\boldsymbol{A}|\boldsymbol{p})P(\boldsymbol{p})} \tag{4.2}$$

$$\approx \frac{P(\boldsymbol{A}|\boldsymbol{W})P(\boldsymbol{W})}{\max_{\boldsymbol{p}} P(\boldsymbol{A}|\boldsymbol{p})P(\boldsymbol{p})} \tag{4.3}$$

$$\approx \frac{P(\boldsymbol{A}|\boldsymbol{W})}{\max_{\boldsymbol{p}} P(\boldsymbol{A}|\boldsymbol{p})} \tag{4.4}$$

但し、$\boldsymbol{p}$は特定の言語で許される音素や音節の系列である。3番目の式は、単語や音素レベルでの確率が1又は0の確定的な言語モデルを仮定した場合である。この式で分母はサブワードモデルによる最適音素／音節系列の尤度と考えることができ、$\hat{P}(\boldsymbol{W}|\boldsymbol{A})$ は尤度比の形となる。本研究で検討する未知語処理法と5章で述べる発話のリジェクションの手法は、原理的に式(4.4)の各尤度の利用に基づいたもので、具体的には $P(\boldsymbol{A}|\boldsymbol{p})$ に相当するモデルとして音節単位モデル(HMM)を用いたものである。

ここで述べる未知語処理の基本的な考え方は、上述のように未知語を任意の音節連鎖と仮定し、連続音声認識において利用する単語レベルの言語制約を図4.1の例のように拡張することである。図に示されるように未知語と冗長語では出現箇所の仮定が異なるので、構文レベルで区別する。Asadiら[56]の提案している方法も同様であるが、未知語に相当するモデルは登録語と同様に明示的に一つの音響モデル（任意の音素の

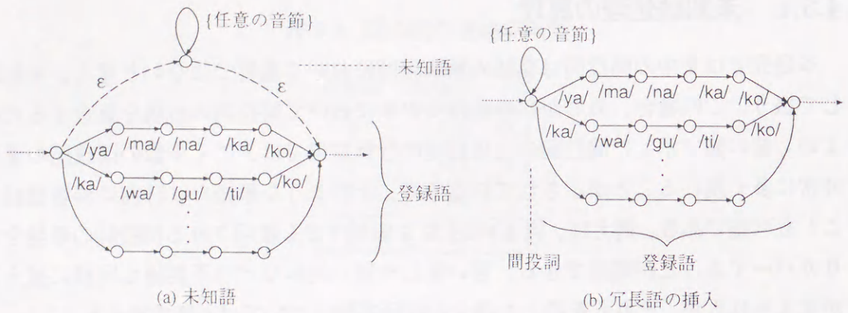


図 4.1: 未知語処理のための言語制約

連鎖を表す HMM）で表されている。ここで述べる方法では、特に未知語の音響的モデルを構成することなく、連続音声認識アルゴリズムに未知語処理を組み込む方法を採っている。この方法の利点は、(1) 未知語が音節列として得られること、(2) 後述のように未知語に対して制約を課すことが容易であること、(3) 計算を近似的に行なうことで未知語処理に要する処理量を大きく節減できること、である。未知語は一般に音節長が未知であり、任意の文節や文節の間において出現し得る。従って、連続音声認識のための探索空間（パープレキシティ）を大きく増加させるため、処理量を抑えることは重要といえる。

4.3.2 連続音声認識における未知語・冗長語処理法

本節では、文脈自由文法の構文規則を用いた連続音声認識システムを例として、未知語処理の実現方法について述べる。認識のための言語的な制約を文脈自由文法で記述する場合、間投詞に関しては、文中にそれらが挿入された場合に文の解析が可能でなければならないので、新たに間投詞カテゴリの書換え規則を文法に追加しておかなければならない。未知語や冗長語は、書換え規則の中で一つの特殊な終端記号（未知語）を用いて記述しておく。例えば図 4.2のような書き換え規則を追加する。間投詞を表す非終端記号 “INTJ” は、更に書き換え規則の任意の位置、特に文頭や文節の境界部分に挿入される。

音声認識の処理では、“未知語” を表す単語仮説を照合する場合に、特別に未知語のモデルに従って音声との照合を行なう。ここでは、未知語のモデルとして任意の音節連鎖を考えるので、3.4.2節 (b) で説明したワードスポッティング法と同様に、O(n)DP 法に基づいて連鎖制約の無い連続音節の認識を行なう。この未知語の照合に用いる音響的モデルは、一般に単語を構成する音響的モデルと同じであるので、そのような場合

文頭の間投詞の挿入のため			湖に関する未知語の扱いのため		
S	→	INTJ S	p-place	→	富士五湖
INTJ	→	intj	p-place	→	山中湖
INTJ	→	ε	p-place	→	河口湖
intj	→	%	p-place	→	%

（“ε” は空記号、“%” は未知語を表す特殊な記号）

図 4.2: 未知語処理のための文法の例

には未知語の照合のスコアは登録されている単語の照合スコアより常に高くなる。そこで、未知語の照合スコアにはペナルティとしてのスコアを与える必要がある。本研究では、未知語のフレーム長又は音節数に比例したペナルティスコアを与えて評価を行なっている。

4.3.3 厳密な未知語・冗長語処理アルゴリズム

未知語を扱うための直接的な方法は、Asadi ら[56]のように、“未知語” の音響的モデルを構成し、一般の単語と同様に “未知語” を辞書登録する方法である。ここで述べる方法は、“未知語” の音響的モデルを構成することなく、従来の音声認識アルゴリズムの変更によって実現するものであり、“未知語” に対するモデルの仮定が同じであれば同じ認識結果が得られる。

従来の連続音声認識アルゴリズムへ追加・変更が必要な処理は、“未知語” 仮説に対する照合の処理である。つまり、“未知語” を照合する場合は、通常の単語と違って決まった音節表記が与えられないので、任意の音節系列を仮定して照合を行なう。そのアルゴリズムとしては、3.4.2節 (b) で用いた O(n)DP 法（One Pass Viterbi 法）を、認識の単位を単語から音節に置き換えることで実現できる。但し、“未知語” 仮説の照合は、仮説が予測されたところの解析の状態 q毎に独立して計算を行なう必要がある。あるひとつの解析の状態にある未知語仮説の照合処理を考えると、3.4.2節 (b) で示した単語レベルのフレーム同期処理はここでは音節単位に置き換えられ、音節境界を仮定した設定の式 (3.8),(3.9) は、未知語仮説の前に先行する部分文仮説の最適な累積尤度を $L_p(t)$ とすると、

$$L^n(t-1,1) = \max\{L_p(t-1),\ L(t-1)\} \tag{4.5}$$

$$B^n(t-1,1) = \begin{cases} t-1 & \text{式 (4.5) の右辺第 1 項を選択時,} \\ B(t-1) & \text{式 (4.5) で右辺第 2 項を選択時} \end{cases} \tag{4.6}$$

となる。ここで、$B(t)$ は時刻 t で終端する未知語仮説に対するバックポインタ（始端位置−1）となる。この未知語処理法によって、結果的に、一つの未知語仮説あたり音

節数のオーダの処理が増えることになる。そこで、次の節でこの方法の近似的な実現法について述べる。

4.3.4 近似的な未知語・冗長語処理アルゴリズム

前述の厳密な未知語処理アルゴリズムでは、未知語仮説の数に応じて処理が増加する。しかし、アルゴリズムから明らかなように、それぞれの未知語仮説の照合のための計算法は全く同じで、それぞれの未知語仮説が現れた部分文におけるコンテキストによって、スコアや開始フレーム位置などが異なるだけである。そこで、3.4.2節と同様にワードスポッティング法に基づく拡張連続DP法の考え方を導入して、未知語仮説の照合を一つにまとめることを考える。以下に、3.4.2節(b)の方式のようにO(n)DP法を用いる場合について述べるが、3.4.2節(c)の方式でも同様に実現できる。

改めて、構文的制約を用いた連続音声認識の処理法について考えてみる。ある構文的な状態qのフレームiにおける累積照合スコア（対数尤度）$L_q(i)$は、

$$L_q(i) = \max_{p,m,n}\{L_p(m) + L^n(m+1:i)\} \tag{4.7}$$

但し、

$$\begin{cases} \delta(p,n) = q(\text{状態}\,p\,\text{から単語}\,n\,\text{を生成して状態}\,q\,\text{に達する}) \\ L^n(m+1:i)\,\text{は単語}\,n\,\text{の}\,m+1 \sim i\,\text{フレームまでの累積照合スコア} \end{cases}$$

となる。

一方、音節連鎖に制約が無い場合は状態数が1つに縮退するので、任意の音節系列の最大累積照合スコア$L(i)$は、

$$L(i) = \max_{m,n}\{L(m) + L^n(m+1:i)\} \tag{4.8}$$

で表され（但し、nは音節）、3.4.2節(b)に示したアルゴリズムを音節単位に置き換えれば$O(n)$の計算量で求められる[37]。ここで、式(4.7)でnが未知語の時、$L^n(m+1:i)$を式(4.8)で求まる値で近似する。つまり、$L(i)$に対応する音節系列の最適状態系列のバックトレースによって得られる音節境界の集合を$\{B(i)\}$とすると、nが未知語のときの式(4.7)の第2項は、$m \in \{B(i)\}$の制約の下では$L(i) - L(m) + C_p \times (i-m)$（但し、$C_p$は未知語仮説に対するペナルティ値）によって求めることができる。なお、式(4.8)で求まる$L(i)$は、5章で述べる発話のリジェクションに利用することも考えられる[66]。

前節および本節では、3つの未知語処理アルゴリズム—(a)厳密な計算による方法、(b) Bundleサーチ型手法に基づく近似処理法、(c) O(n)DP法に基づく近似処理法—の

実現について述べた。(a)の方法は、Asadiらの方法[56]と同様に未知語モデルを仮定した最適な連続音声認識の探索を行なうものであるが、明示的な未知語モデルを用いない点が異なる。北らの方法も同様であるが、文節毎に発声した音声を対象としている[63]。(b)と同様な方法として、伊藤らの連続音声認識システム[64]では、音韻を照合単位として未知語処理を実現しており、日本語の言語音声として不適格な音韻連鎖を排除するために、音韻連鎖の統計モデルを併用している。これらの方法とは別に提案する(c)の方法は、音節単位での連鎖制約のない照合を行なうもので、未知語照合のスコアの算出が容易であり、応用ではOne Pass法以外の認識アルゴリズムへの適用も容易である[3]。また、次の章で述べるような発話全体のリジェクションへの適用が可能である。これらの未知語処理のアルゴリズムに必要な主な処理を比較すると、厳密なアルゴリズムの場合は生成される未知語仮説の数に処理量が比例するのに対して、近似的なアルゴリズムでは基本的に一つの未知語仮説に対する照合に相当する式(4.8)の計算だけなので、ほぼ生成される未知語仮説数の比だけ改善の効果がある。近似的なアルゴリズムによって計算結果に影響する点は、未知語仮説の開始フレームの近似によるものであるが、これは音節モデルによるセグメンテーションの精度に依存することになる。

4.3.5 未知語・冗長語仮説に対する制約

前述の未知語処理法は、言語的制約を用いないでサブワード単位の音響的モデルによる認識を行なうが、未知語の検出精度を良くするために何らかの制約を与えることが望まれる。これまでには、未知語に対する言語的制約として、音韻レベルの連鎖制約（bigramやtrigram）の利用や[63, 64]、意味的に類似した単語間での発音の類似度の利用[76]などが検討されている。本研究では、音節単位のモデルを利用しているため、音韻レベルの連鎖制約はかなり含まれているといえる。音節レベルの連鎖制約も考えられるが、未知語の場合には一般に強い制約にはなり得ない。後者のような、発音の類似による制約も有効と考えられるが、語彙に依存した制約が必要になる。ここでは、ヒューリスティックな2通りの制約を考える。

- 未知語の長さ（音節数）の制限
- 一発話中の未知語の出現を一つに制限

1番目の長さの制約は、一音節に相当する短い音声区間が未知語として過剰検出されることを防ぐのに特に有効と考えられる。一方、2番目の未知語の出現数の制約は、発話

[3]第6章で用いた2種類の文節スポッティングベースのシステムへの適用も行なっている。

の長さが比較的短い情報検索のようなタスク、語彙が限定され易いタスク、出現頻度の高い冗長語を辞書登録している場合のその他の冗長語の処理、などにおいて有効に利用できるものと考えられる。未知語を二つ以上含む発話は、たとえ正しく同定できても言語理解が困難なため、そのような場合がないように辞書を大きくする（例えば数千語以上）などが必要であり、ここでは処理対象外とする。但し、システムの頑健性を維持するためには、このような文に対してはリジェクションを行なうことにより対処できる（5章で述べる。）。

未知語の音節数の制限は、前述の未知語処理のアルゴリズムの一部変更で容易に実現できる。未知語の出現数の制限は、Viterbi 法に基づく構文制御型の連続音声認識アルゴリズムを用いる場合、状態毎のローカルな単語尤度計算のときに、未知語を一つだけ含む仮説と一つも含まない仮説の2種類の累積尤度を保存するための記憶を用意し、未知語の照合を始めるときの累積尤度の初期値を後者の仮説の累積尤度を用いるようにすることで実現できる。結果として、未知語を一つ含んだ認識結果と未知語を含まない（未知語検出を行なわないのと同じ）認識結果の2種類が得られる。ここで、スコアに基づいてその2種類のうちの一つを最終認識結果とすればよいが、更に意味知識など top-down の知識を利用して未知語の有無が妥当であるか判断することも可能である。

4.4 未知語・冗長語を含む発話による認識実験

4.4.1 評価用システム - SPOJUS-SYNO Y -

本研究では、未知語処理の有効性を調べるために、One Pass 法に基づくシステム SPOJUS-SYNO-X に対して未知語処理を追加したシステムを新たに構築した。図 4.3 に、新たな評価用のシステム SPOJUS-SYNO-Y の概略を示す。

本システムは、3章で認識実験に用いた SPOJUS-SYNO-X に対して、音声の分析条件と音響モデルが若干変更されている。音声の分析と特徴ベクトルへの変換は、表 4.5 の条件で行なう。認識に用いる HMM は、カテゴリ数が 113 個のコンテキスト独立の音節単位のモデルである。HMM の構造は3章におけるシステムと同様、5 状態 4 出力分布で、出力分布は単一のガウス分布を仮定し、状態毎に離散継続時間分布（最大=14 フレーム）を持っている。実験では、継続時間制御の効果を高めるため、離散継続時間分布を 3 乗して重み付けしている。

評価用システムは、"富士山観光案内" の音声対話システムにおけるユーザ発話を認識することを想定している。使用する文脈自由文法には、対話システムにおける次発

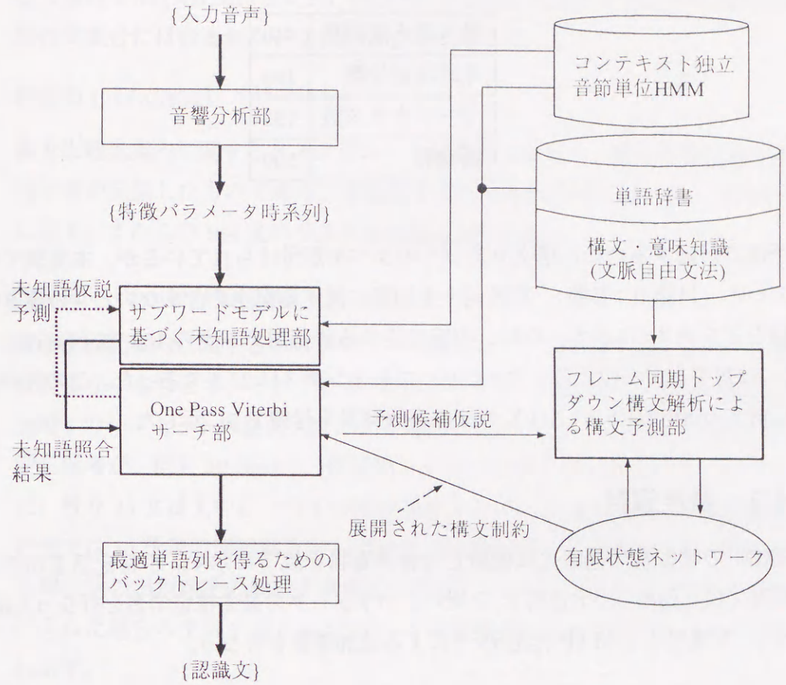


図 4.3: SPOJUS-SYNO-Y の概略図

表 4.5: 音声の分析条件

サンプリング周波数	12 kHz
フレーム長	21.33 msec. (256 points)
フレーム周期	8 msec. (96 points)
プリエンファシス	$1 - 0.98Z^{-1}$
分析窓	Hamming Window
分析	14 次の LPC 分析
特徴パラメータ	10 次メルケプストラム係数 + 10 次回帰係数

表 4.6: 文脈自由文法の規則数（富士山観光案内タスク）

書き換え規則数	440
非終端記号数	184
ワードクラス数	193
語彙数	500

話予測に対応するために構文カテゴリのラベルが付けられているが、本実験では無視している（付録B.1参照）。間投詞や未知語に関する記述を含まないときの書き換え規則数などを表4.6に示す。なお、次節で述べる評価用データの104文（Fuji104：付録B.3）に対する文法のテストセットパープレキシティは29.3であった。この評価用データに対するSPOJUS-SYNO-Xの認識実験結果を付録B.6に示した。

4.4.2 音声資料

HMMの学習及び適応化に使用した音声資料を以下に示す。不特定話者用のHMMの学習では、始めに“学習用1”を用いてパラメータの最尤推定学習を行なった後、“学習用2”を使用してMAP推定法[75]による追加学習を行なう。

- 学習用1

  ATR研究用日本語音声データベース中の、連続発声文データ（A～Jセット、計503文）を男性6名が発話した音声の音節データを用いた。ただし、不足する音節カテゴリに対しては、ATRデータベース中の、音韻バランスのとれた216単語を男性10名が発話した音声の音節データを用いている。

- 学習用2

  日本音響学会研究用連続音声データベースVOL 1～3を用いて、基本モデルから標準モデル（113音節の不特定話者HMMモデル）を学習した。話者数は男性30名（1名あたり約150文）で、合計約4500文である。

話者適応化と評価実験のために、以下の音声資料を使用した。認識実験では、あらかじめ“話者適応化用”を使用して、MAP推定法によって不特定話者モデルに対する話者適応化を行なう。評価実験は、未知語を含む発話と冗長語（間投詞）を含む発話に分けて行なうため、2種類を用意している。

- 話者適応化用（Fuji20）

富士山観光案内に関するタスク[51]に依存したユーザ側発話20文を、静かな録音室で男性6名(AK,MM,SA,TK,TS,YM)が発話したものである[83]。以下の評価用の文集合には含まれない文だけからなっている。

- 評価用1（Fuji104）

  富士山観光案内に関するタスクのユーザ側発話104文で、静かな録音室で同じ男性6名が発話したものである。未知語を含む発話の評価に用いる。文例を図4.4に示す。またこの104文のリストを付録B.3に示す。

- 評価用2（Fuji50, Fuji_k50）

  富士山観光案内に関するタスクのユーザ側発話で、間投詞を含むものと、間投詞を含まないものとで、それぞれ50発話からなる。間投詞を含まない発話は、評価用1のサブセットの話者AKによる発話である(Fuji50)。間投詞を含むものは、同じ話者が、同じ50文の中に間投詞を入れて発声し直したもので、25文は文頭に、残り25文は文中に一つずつ間投詞が入っている(Fuji_k50)[4]。間投詞入りの評価文は10種類の間投詞を含んでおり、4.2節で示した人間どうしの対話における間投詞の出現頻度の統計を考慮して作成している[5]。間投詞を文頭または文中に入れた場合の文例を図4.5に示す。また間投詞入り評価文のリストを付録B.5に示す。

4.4.3 未知語を含む発話に対する評価実験

この認識実験では“評価用1”の発話データを使用する。前述のシステムの辞書には“評価用1”に含まれる単語が全て登録されているので、未知語を含めるために一部の単語をシステムの辞書から取り除いて評価する[6]。以下に2種類の実験結果を示すが、削除した単語は異なっている。

(a) 未知語処理法の比較

4.3.3節と4.3.4節で述べた3種類の未知語処理アルゴリズムによる評価実験について述べる。システムの辞書からは、観光場所名に関する2つのワードクラス（“koyu_mountain”と“koyu_place”）に含まれる全ての単語を削除し[7]、それらのワードクラスに未知語

[4]間投詞を含む発話のみ、適応化発話データと音声の収録時期が約10カ月離れている。
[5]付録B.4を参照。頻度順の上位10種類の間投詞を使用。
[6]実際には、ワードクラス内の単語のエントリを削除する。
[7]“koyu_mountain”の方は一つの単語（“富士山”）しか含まない。

001: 富士山周辺で特に観光というと、何 (naN) か有りますかね。
002: 富士五湖というのはだいたいどちら側にあるんですか。
003: 富士五湖っていうと五つの湖ですよね。
004: 富士五湖にはどんな観光地があるんですか。
005: そうですか。山中湖の方はどうですかね。
006: 遊覧船はだいたい何分ぐらいでまわれるんですかね。

図 4.4: 認識評価用 Fuji104 の文例

001: [えーと] 富士山周辺で特に観光というと、何 (naN) か有りますかね。
013: [あの] 富士山に登山したいと思うんですけども。
042: [あのー] 風穴は一般公開されているんですか。
050: [あ] じゃ具体的に決まったら、またそちらに連絡したいと思います。
055: どういったところが [あの] 見所でしょうか。
057: ですが、[えーと] 頂上までは行かない予定です。
067: ペンションとかは [あの] ないんでしょうか。
094: どういった [ま] 宿泊施設がありますか。

図 4.5: 間投詞を含んだ認識評価用の文例 ([ ] の中が間投詞)

のエントリーを追加した。削除された単語は 20 個の固有名詞であるが、評価文に含まれるのは { 五合目, 富士五湖, 富士山, 河口湖, 西湖, 山中湖 } の 6 単語だけである。これらの未知語を含んでいる評価文は、104 文中で 21 文あった。表 4.7に、それぞれの未知語処理法による文認識率を示す。但し、文認識率は、発話文の未知語を除く全単語が正しく認識されている文の割合を表している。この表で、"最適型" は 4.3.3節のアルゴリズム、"O(n)DP 法" と "Bundle 法" は 4.3.4節で述べた近似的な方法に対応している。結果を見ると、認識率では O(n)DP 法が最適法と同等であり、近似的な方法による顕著な差はあまりないと考えられる。一方で、計算量は近似的な方法が最適法よりも 3 割近く少なくなるのが分かる。近似的な方法では、未知語処理のために費やされる計算のうち、最も多い Viterbi 計算の処理量はほぼ音節カテゴリ数にのみ依存する。従って、本研究のように音節カテゴリの音響モデルを未知語処理に用いる場合には、数十単語の語彙の追加程度[8]の計算量の増加と、その他の未知語処理に関する若干のオーバヘッド程度で抑えることができる。この近似的な方法による計算量増加の抑制の効果は、未知語が多くのワードクラスで登録される場合にはより顕著になるはずである。

[8] 音節カテゴリ数が 110 個であれば、一単語が 3 音節として約 37 単語分、4 音節として 28 単語分である。

表 4.7: 未知語処理法の比較 (文認識率 [%])

未知語区間のペナルティスコア = −3/ frame

評価文	最適法	Bundle 法	O(n)DP 法
未知語あり (21 文)	66.7	61.9	66.7
未知語なし (83 文)	78.3	79.5	78.3
認識時間 (秒)	284	202	212

使用計算機: OMRON LUNA-88k (25MIPS)

(b) 未知語の制約を用いた評価実験

前の実験では、2 つのワードクラスから全ての単語を削除して未知語としたが、一般に同一のワードクラス内に発音や単語表記が似ている単語が多いので、未知語検出の条件としては比較的易しいと考えられる。そこで、この実験では 5 種類のワードクラスから 10 種類の単語だけを削除した。表 4.8に削除した単語とその単語が属していたワードクラス名を示す。削除した単語が属していたワードクラスは、特に観光地や乗り物などの名詞の単語集合を表している。実験は、未知語が属していた 5 種類のワードクラスに対して "未知語" のエントリーを追加して行なった。この辞書の変更によって、"評価用 1" の 104 文のうち 18 文が未知語を一つづつ含むようになった。実験結果を表 4.9、4.10に示す。ここでは O(n)DP 法に基づく近似的アルゴリズムによる結果だけを示す。本実験では、未知語区間のペナルティスコアとして、未知語仮説の音節数に比例して実験的に定めた-40 という値を累積対数確率スコアに加えた。前述のようにフレーム単位でペナルティスコアを与える場合と比べて、実験的には性能の差がほとんどなかったが、ペナルティスコアの違いや話者の違いによる性能の変動がやや少ない効果が見られた。

前の節と同様に、文認識率は発話文の未知語を除く全単語が正しく認識されている文の割合を表している。また、未知語の制約の "L" は未知語の長さを 2~10 音節に制限した場合、"N" は一文中の未知語の出現数を一つに制限した場合を示している。話者平均の結果をみると、未知語処理による未知語を含まない発話の認識率の低下は約 2%であるが、未知語を含む発話に対して、未知語が無い場合における正解認識文の 7 割以上が正しく認識できている。また、どちらの未知語の制約についても、未知語検出に有効に働いているといえる。一部の話者 (特に、MM と SA) は未知語を含む文の認識率が特に低いが、これらの話者は未知語なしで未知語処理をしない場合の認識精度も悪い。このことから、元の認識精度と未知語の検出率との間に相関があると考え

表 4.8: 削除した単語リスト

ワードクラス名	削除した単語
koyu_place	山中湖
	五合目
norm_build	宿泊施設
	観覧車
	遊覧船
norm_spot	観光地
	温泉
norm_place	頂上
vehicle	バス
	電車
	観覧車
	遊覧船

られる[9]。未知語を含む発話の認識誤りを調べると、ほとんどは未知語付近でのセグメンテーション誤りに起因しており、特に短い接続詞や感動詞が未知語の一部分となった誤りが多かった。また、上述の2種類の未知語の制約を用いない場合は、一つの未知語の発声区間が助詞を挟んで2つの未知語として認識されるような場合が目立ったが、どちらの未知語の制約もこの種の誤認識の改善に有効であった。

4.4.4 間投詞を含む発話に対する評価実験

前述の未知語処理を適用し、間投詞を未知語として処理する場合の有効性を調べる。前述の間投詞に関する分析結果から考えると、前述の未知語の制約条件は、間投詞を全て未知語として扱う場合には条件が厳しいと考えられるので、2種類の制約ともこの実験では用いない[10]。実験では、出現する間投詞を単語登録した場合との比較も行なった。どちらの場合も、文頭や接続詞の後ろ、助詞の後ろなどに間投詞が出現可能なように間投詞の非終端記号を文法に組み込んでいる。間投詞の単語登録を行なう方法では、評価文中に現れない間投詞を最大で78個（4.2節の間投詞の調査において抽出されたもの[69]）追加登録した場合も検討した。78個の間投詞のリストは付録B.4に示す。

図4.6に未知語処理のみによる認識結果の例を示す。認識結果の評価として、図(a)

[9]この関係については5章で詳しく検討している。

[10]但し、実際には、大部分の間投詞を登録して十分大きな辞書を持っていると仮定すれば、この制約を用いることも可能と考えられる。

表 4.9: 未知語 (10 単語) を含む 18 文の文認識率
（括弧内は正解認識文数）

条件	未知語の制約	話者						
		AK	MM	SA	TK	TS	YM	ALL
未知語なし& 未知語処理なし	—	72.2 (13)	50.0 (9)	44.4 (8)	83.3 (15)	83.3 (15)	83.3 (15)	69.4 (75)
未知語あり& 未知語処理あり	なし	50.0	33.3	22.2	55.6	55.6	50.0	44.4
	L	50.0	33.3	27.8	61.1	66.7	55.6	49.1
	N	55.6	38.9	27.8	61.1	66.7	55.6	50.9
	L,N	55.6	38.9	27.8	66.7	72.2	55.6	52.8
		“未知語なし&未知語処理なし”での正解文のみを対象						
未知語あり& 未知語処理あり	なし	69.2 (9)	55.6 (5)	50.0 (4)	66.7 (10)	66.7 (10)	60.0 (9)	62.7 (47)
	L	69.2 (9)	55.6 (5)	62.5 (5)	73.3 (11)	80.0 (12)	66.7 (10)	69.3 (52)
	N	76.9 (10)	66.7 (6)	62.5 (5)	73.3 (11)	80.0 (12)	66.7 (10)	72.0 (54)
	L,N	76.9 (10)	66.7 (6)	62.5 (5)	80.0 (12)	86.7 (13)	66.7 (10)	74.7 (56)

表 4.10: 未知語 (10 単語) を含まない 86 文の文認識率
（括弧内は正解認識文数）

条件	未知語の制約	話者						
		AK	MM	SA	TK	TS	YM	ALL
未知語なし& 未知語処理なし	—	87.2 (75)	76.7 (66)	65.1 (56)	76.7 (66)	83.7 (72)	83.7 (72)	78.9 (407)
未知語あり& 未知語処理あり	なし	83.7	74.4	62.8	73.3	82.6	76.7	75.6
	L	84.9	75.6	64.0	75.6	82.6	77.9	76.7
	N	83.7	75.6	62.8	73.3	82.6	76.7	75.8
	L,N	84.9	75.6	64.0	75.6	82.6	77.9	76.7
		未知語過剰検出の個数						
未知語あり& 未知語処理あり	なし	5	6	6	6	6	8	37
	L	3	3	3	3	6	6	24
	N	5	3	4	6	5	8	31
	L,N	3	2	3	3	5	6	22

文番号
002 : /peto/ 富士五湖 という の は だいたい どちら 側 に ある んですか
(入力文: [えーと] 富士五湖というのはだいたいどちら側にあるんですか。)
008 : /waNnoo/ 富士急ハイランド は 結構 大きい んです かね
(入力文: [あの] 富士急ハイランドは、結構大きいんですかね。)
060 : /ko/ サファリパーク というの は /pedetoo/ どんな 車 でも 入れるんです か
(入力文: サファリパークというのは[えーと] どんな車でも入れるんですか。)
084 : それら は /ma/ 十分 見て まわれる んでしょう か
(入力文: それらは[まー] 十分見てまわれるんでしょうか。)

(a) 間投詞以外が正しく認識された例

文番号
003 : え、 /too/ 富士五湖 っていう と 五つ の 湖 です よね
(入力文: [えーと] 富士五湖っていうと五つの湖ですよね。)
004 : /ne/ え、 /to/ 富士五湖 には どんな 観光地 が ある んです か
(入力文: [えーと] 富士五湖にはどんな観光地があるんですか。)
015 : え、 で、 どの 辺 まで 車 で 登山 できる んでしょう か
(入力文: [えー] どの辺まで車で登山できるんでしょうか。)
078 : はいはい、 よろしく おねがいします
(入力文: はい、[あ] よろしくおねがいします。)
081 : /gi/ え、 /paku/ 電車 で 行こう かと 思っています
(入力文: いえ、[あ] 電車で行こうかと思っています。)

(b) 文頭の間投詞が接続詞、感動詞として誤認識された例

文番号
071 : 今 からは /kenei/ 予約 しても 間に合う でしょう か
(入力文: 今から[えー] 予約しても間に合うでしょうか。)
074 : 頂上 の 方 へ /nepe/ 料金 なら /ide/ 結構 です けど
(入力文: 先ほどの[えー] 料金ぐらいでけっこうですけど。)
088 : 新富士 までは /Nnoobo/ 値段 は どれ くらい かかります か
(入力文: 新富士(siNfuji) まで[あのー] 値段はどれくらいかかりますか。)
096 : じゃ、 /a/ 車 か ペンション なら その 近く に ある んです ね
(入力文: じゃあ、[ま] ペンションなら、その近くにあるんですね。)

(c) 間投詞以外の部分でも誤認識となった例

図 4.6: 間投詞を含む発話の認識結果例

表 4.11: 間投詞を含む発話の認識性能
(間投詞の間違い/挿入を無視した文認識率. 括弧内の値は、更に間投詞と文頭の接続詞、感動詞の間違いを無視した場合.)

＼評価項目 / 文法・辞書＼	文認識率(%) 間投詞なし	文認識率(%) 間投詞あり	予測仮説数 (一文当たり)	照合単語数 (一文当たり)	認識時間 (秒)
間投詞なし文法	86.0	4.0 (32.0)	1445	267	247
10 間投詞登録†	88.0	74.0 (76.0)	1909	321	263
30 間投詞登録†	86.0	68.0 (74.0)	3011	494	337
78 間投詞登録†	84.0	68.0 (76.0)	5638	827	597
未知語処理法‡	82.0	56.0 (72.0)	1273	391	331
未知語処理 +10 間投詞登録 †	86.0	70.0 (74.0)	2123	523	391
未知語処理 +文頭 10 間投詞登録 †	—	70.0 (72.0)	1560	404	329

(†beam search 幅= 45, ‡beam search 幅= 20
認識時間は、OMRON 社製 LUNA-88k(25 MIPS) での CPU 時間で計測)

のように間投詞の間違いや挿入などを無視した場合の単語列の正解認識率を求めた。表4.11に両手法による実験結果を示す。未知語処理による結果は、間投詞を登録する方法よりも文認識率が劣っているが、図4.6(b) の例のような間投詞と文頭の接続詞、感動詞の違いを無視した場合の文認識率（表の括弧内の値）は同等であった。また、間投詞登録と未知語処理を併用した場合は、文認識率も向上している。表の最下段は、文頭や接続詞の後のみ間投詞を単語登録した場合の結果であるが、全ての文節間で登録された間投詞を許した場合と同等な性能が得られ、計算量も16%程度減少している。間投詞の登録数の増加では認識性能はほとんど低下していないが、計算量の増加が大きいため、現実的には未知語処理と主な間投詞登録の併用が有用といえる。

4.5 まとめ

本章では、未知語や間投詞を音声認識で扱うためのアプローチについて検討した。

まず、サブワード単位の音響的モデルを用いた未知語処理法について検討した。近似的なアルゴリズムを考えたが、厳密な方法と同等の精度が得ることができ、未知語処理の計算量を大きく節減できることが実験的に確かめられた。また、2 種類の未知語の仮説に対する制約の効果を調べた結果、どちらの制約も未知語検出精度の向上と

未知語の過剰検出の削減に効果があることが分かった。

次に、間投詞を未知語として扱うことを検討し、間投詞を辞書登録する場合と比べたときの有効性について調べた。実験の結果から、間投詞を未知語として扱う場合には、文頭の間投詞の発声部分で接続詞や感動詞などが誤って認識されることが多いことが分かったが、それらを無視した場合には同等の認識精度が得られることが分かった。また、文頭に接続詞などと間違いやすい間投詞の一部を登録して未知語処理と併用する場合には、認識精度が改善され、間投詞の登録を増やした場合に比べて処理時間もかなり少なくて済むことが分かった。

現在の方法は、音響的なモデルの尤度だけで未知語としての仮説のスコアを求めるため、発音のスペクトルのゆらぎやサブワードモデルの精度の問題のために一部のサブワードモデルの尤度が低くなるような場合に、十分な精度が得られていないと考えられる。更に大語彙の場合でも信頼性のある未知語処理を実現するには、より多くの知識源を考慮して仮説の信頼性の尺度を求めるような方法が必要になってくると考えられる[77]。

第5章

サブワードモデルを用いた発話のリジェクション

5.1 はじめに

音声認識システムにおける発話のリジェクション (rejection) の機能は、ユーザの発話内容に対して柔軟に対応できるユーザインタフェースを実現するために特に重要な機能の一つである。本章は、4章で述べた未知語処理の考え方を発話のリジェクション機能の実現法として捉え、その有効性を理論的および実験的に検討している。

本研究のようにサブワード単位の音響的モデルを用いる未知語処理の考え方は、スポッティングでのキーワード以外 (non-keywords) の棄却や[57, 58]、文発話の棄却[66]などにおいても採用されている。このアプローチの有効性は音響的モデルの精度に依存するといえるが、具体的にシステムの認識精度と未知語の検出精度または棄却率との関係について論じられたことはなかった（実験的な評価では文献[78, 79]の報告がある）。

本研究では、まず前述の未知語処理の考え方を孤立単語認識に適用した場合を想定し、シミュレーション法によって単語認識率と棄却率の関係などを調べた。また、このシミュレーション結果を検証するため、孤立単語認識システムによる認識実験を行ない、同様な検討を行なった。ここでのシミュレーションによる分析は、孤立単語認識の場合でのみ考える。その理由は、文認識の未知語処理の場合には、システムが用いる言語的知識や認識アルゴリズムなどの要素の影響を無視できないために、理論的考察は勿論、シミュレーション法による分析は困難と考えられるためである。しかし、単語認識率、パープレキシティと文認識率の近似的な関係が示されているように[80, 82]、単語認識での性能に基づいて文認識の性能をある程度推測することは可能であろう。この章の最後に示す文音声認識での評価は、4章の実験で使用した連続音声認識システム

SPOJUS-SYNO-Y によるいくつかの実験に基づいて行なった。

5.2 シミュレーションによるリジェクション性能の推定

5.2.1 未知語検出法の仮定

孤立単語の音声認識システムでは、未知語の発声に対する認識結果を棄却(reject)する簡単な方法として、あらかじめ定めた認識スコアのしきい値で判定する方法がある。しかし、認識の結果得られる単語の尤度スコアは、人や発話環境の違いによる影響を受けるため、一般に安定したリジェクションは行なえない。そこで、4章で述べたようにgarbageモデルやサブワードの連鎖モデルなどを未知語（background）モデルと仮定して並用する方法が用いられる。4.3.1節で述べたように、未知語モデルの尤度スコアとしてサブワード連鎖（音節／音素連鎖）による尤度を仮定すると、次式の尤度比に基づいて棄却の判定を行なうことが考えられる。

$$\frac{P(\boldsymbol{A}|\boldsymbol{W})}{max_{\boldsymbol{S}} P(\boldsymbol{A}|\boldsymbol{s})} < \theta \tag{5.1}$$

または、

$$\log P(\boldsymbol{A}|\boldsymbol{W}) - \log max_{\boldsymbol{S}} P(\boldsymbol{A}|\boldsymbol{s}) < \theta' \tag{5.2}$$

ここで、一般に $P(\boldsymbol{A}|\boldsymbol{W})$ は単語や文単位の音響的モデルの尤度を表し、$P(\boldsymbol{A}|\boldsymbol{s})$ サブワード系列の音響的モデルの尤度を表している。また、しきい値θは、未知語の棄却率(correct rejection rate)と登録単語の発声が誤って棄却される割合(false rejection rate)の間でのtrade-offを決定するものである。本研究では式(5.2)のようにサブワードモデル（特に音節）を用いた未知語モデルを考える。これまで、音韻認識率と単語認識率との関係などについては検討されているが[81, 80]、単語認識率と上述のリジェクションの方法による棄却率との関係は明らかでない。次節では、音節単位と単語との間の認識精度の相関を仮定して未知語の検出（棄却）のシミュレーションを行い、リジェクションの性能を評価する方法について述べる。

5.2.2 シミュレーション法

孤立単語認識における未知語検出性能を推定するために、単語認識器は、単語カテゴリー数と認識率の関係の評価法[80]に基づいてシミュレーションを行なう。同時に、前述の音節単位の任意の連結による認識を併用した未知語検出法のシミュレーションを行ない、単語認識率と未知語検出率の関係を求める。

単語認識器のシミュレーションでは、次のようなパラメータを考える。

- 単語カテゴリ数： M
  - 正解単語カテゴリのスコア分布： $N(\mu_1, \sigma_1^2)$
  - 誤り単語カテゴリのスコア分布： $N(\mu_2, \sigma_2^2)$
- 音節カテゴリ数： N
  - 一単語当りの音節数： L
  - 正解音節カテゴリのスコア分布： $N(\frac{\mu_1}{L}, \frac{\sigma_1^2}{L})$
  - 誤り音節カテゴリのスコア分布： $N(\frac{\mu_2}{L}, \frac{\sigma_2^2}{L})$

但し、$N(\mu, \sigma^2)$ は平均値μ、分散σ^2の正規分布である。この単語認識器の仮定においては、単語又は音節どうしの音響的な類似性とは無関係に、カテゴリ数Mによって認識性能が求められる。また、音節単位のスコア分布が仮定されているため式(5.2)による評価が行なえる。但し、音節カテゴリに対するスコアの分布は、単語が全て長さL音節で構成されると仮定し、正解単語以外の単語にも正解単語と同じ音節が含まれ得ることを無視している。したがって、音節カテゴリでの認識率は実際より低く推定されるという問題があり、実際には、誤り単語スコアは $N(\frac{\mu_1}{L}, \frac{\sigma_1^2}{L})$ と $N(\frac{\mu_2}{L}, \frac{\sigma_2^2}{L})$ の分布からのスコアの和として求める必要がある。これに関しては次節で述べる。

上述のスコア分布の仮定によって、式(5.2)の規準でのリジェクションの性能が定まる。しかし、上述の仮定においては、計算式でリジェクション性能の理論的な値を求めることは困難であるので、上述のスコア分布に従う乱数の生成に基づいて単語認識と未知語検出のシミュレーションを行う。認識スコアの生成は、未知語の発声（入力）を仮定した場合と、登録語の発声を仮定した場合についてそれぞれ繰り返す。その結果、様々なしきい値での未知語の検出率と、単語認識率 、及び誤棄却率（false rejection rate; 登録語の発声を棄却した割合）の関係が求められる。なお、計算式による近似的な算出方法を付録C.1に示した。

図5.1は、一単語の音声の入力に対してシミュレーションのために生成する認識スコア（四角の中の値）と、未知語検出のための未知語モデルのスコアの求め方の例を示したものである。図に示されるように、正解単語の認識スコアは、L個の正解音節カテゴリのスコアの和として求め、未知語としてのスコアは、各カラム（単語中の音節の位置に対応）毎の正解音節と誤り音節スコアの最大値の和によって求める。一回の試行で求める誤り単語スコアの数は、未知語を仮定した場合は語彙数Mに等しく、未知語を仮定しない場合は正解カテゴリを除く$M-1$個になる。誤り単語スコアは、音節スコアとは独立に $N(\mu_2, \sigma_2^2)$ で求めるか、後述の方法で誤り音節のパラメータを変更した場合は、音節のスコア分布から求める。

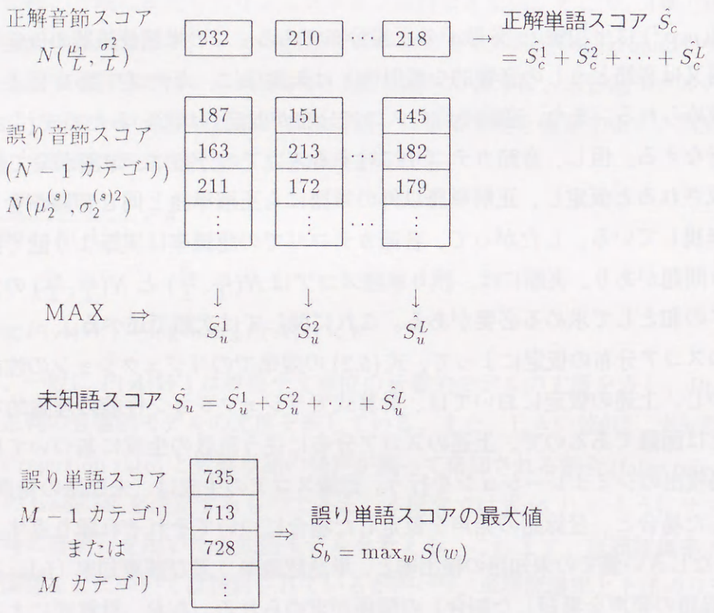


図 5.1: シミュレーションによる認識スコア生成法

5.2.3 シミュレーションの仮定の修正

前述までのシミュレーションでは、音節単位のスコア分布を、単語のスコア分布および単語の音節長 L のみに依存して決めている。しかし、特に誤り単語のスコア分布の仮定については、登録単語の集合によって、実際の誤り音節単位のスコア分布から導出する仮定とのずれが大きくなることが予想される。考えられる主な問題点としては、

- 誤り単語と音節のスコア分布は、誤り単語が全て誤り音節から構成されると仮定している。(入力音節は正解単語以外の単語には含まれないと仮定)
- 従って、実際の登録単語では全ての音節が均等に使われていないため、誤り単語のスコア分布には正解音節のスコアが含まれることを考慮しなければならない。

そこで、シミュレーションの仮定をより改善するために、以下の方法を考える。

- 方法 1

  実際の登録単語では音節が均等に使われていないことが多いので、音節単位のパープレキシティや等価音節出現数（音節の出現頻度を考慮）を求めて、シミュレーションの音節数を補正する。

  一つの近似法として、誤り単語が発声された正しい音節を含む割合が P_{cc} であると仮定する。この場合、誤り単語のスコアの生成では、2 種類の音節のスコアの分布を P_{cc} の割合で切替えて求める。音節のスコア分布はあらかじめ求めておく。P_{cc} は、実際のシステムが持っている辞書に対して、次式で概算する。

  $$P_{cc} = 1 - \frac{1}{L \cdot M \cdot (M-1)} \sum_{w_i} \sum_{w_j, w_j \neq w_i} d(w_i, w_j)$$

  但し、$d(w_i, w_j)$： 2 つの単語 w_i と w_j の、対応する音節シンボル系列間に対するハミング距離

- 方法 2

  誤り単語には正解単語と音響的に近い音節や同じ音節を多く含むものと考えられる。そこで簡単のために、登録単語中の音節の一致の割合を考慮し、以下の計算で誤り音節のスコア分布のパラメータ（$\mu_2^{(s)}, \sigma_2^{(s)2}$）を求める。

  1. 全ての登録単語の対 w_i, w_j についての音節単位のハミング距離の平均を求める。

     $$\begin{aligned} D_h &= \frac{1}{M(M-1)} \sum_{w_i} \sum_{w_j, w_j \neq w_i} d(w_i, w_j) \\ &= L(1 - P_{cc}) \end{aligned}$$

2. 誤り単語は平均的に D_h 個の誤り音節と $(L-D_h)$ 個の正解音節を含むと仮定して、誤り音節のスコア分布のパラメータを求める。単純に、パラメータが次の関係があると仮定する。

$$\mu_2 = \mu_1^{(s)} \cdot (L-D_h) + \mu_2^{(s)} \cdot D_h$$

$$\sigma_2^2 = \sigma_1^{(s)2} \cdot (L-D_h) + \sigma_2^{(s)2} \cdot D_h$$

すると、誤り音節のパラメータは次式で与えられる。

$$\mu_2^{(s)} = \frac{\mu_2 - (L-D_h)\mu_1/L}{D_h}$$

$$\sigma_2^{(s)2} = \frac{\sigma_2^2 - (L-D_h)\sigma_1^2/L}{D_h}$$

- 方法3

前述の問題点より、誤り音節のスコア分布の平均値は最初に述べた仮定よりも幾らか低いと考えるのが妥当である。そこで、音節のスコア分布によるシミュレーションで求まる音節認識率が、実際のシステムと大体合うように誤り音節のスコア分布の平均値を補正する。

$$\mu_2^{(s)} = \frac{\mu_2}{L} \implies \mu_2^{(s)\prime} = \frac{\mu_2}{L} - C_0$$

但し、C_0: 補正値

これらの方法はどれも実際のシステムを近似しているだけなので、方法による違いは若干みられたが、全体の傾向は同じで、異なる結論を導く程は大きくなかった。次節で示すシミュレーション結果は、パラメータを直接修正する必要が無い方法1を採用している（$P_{cc} = \frac{1}{20}$： 5.3節の実験に用いたシステムの辞書から推定された値)。

5.2.4 認識率、リジェクション率、過剰検出率の関係

以前にいくつかの認識率が異なるシステムを仮定して、単語カテゴリー数と単語認識率の関係が求められている[1][80]。そこで表5.1の条件は固定して、代表的な単語認識率と語彙数の組合せに対応するスコア分布のパラメータ集合を決定した。

上記の予備的な実験で求まったパラメータから、5.2.3節の方法1の仮定で用いる音節カテゴリーのスコアのパラメータは次のようになる。

[1]そのような関係は、$\theta = -\infty$ で既知単語を入力した場合のシミュレーション結果に相当する。

表 5.1: シミュレーションの条件

単語カテゴリ数		$10 \sim 1000$
音節カテゴリ数		100
一単語当りの音節数		4
単語認識器の固定システムパラメータ	正解単語カテゴリ	$\mu_1 = 850$ $\sigma_1 = 25$
	誤り単語カテゴリ	$\sigma_2 = 40$

$$\left.\begin{aligned}\mu_1^{(s)} &= \tfrac{\mu_1}{L} = 212.5\\ \sigma_1^{(s)} &= \sqrt{\tfrac{\sigma_1^2}{L}} = 12.5\end{aligned}\right\} \cdots \text{正解音節スコア分布用}$$

$$\sigma_2^{(s)} = \sqrt{\tfrac{\sigma_2^2}{L}} = 20.0 \quad \cdots \text{誤り音節スコア分布用}$$

方法1の場合には、他の2つのパラメータ（$\mu_2^{(s)}$ と P_{cc}）は、システムの性能を制御するものである。図5.2に、単語カテゴリー数と単語認識率の関係を示す。各々の曲線は $\mu_2^{(s)}$ をパラメータとしたもので、システムの認識精度の違いに対応する。この図は、以前の結果[80]とは若干異なるが、これは方法1を用いることで単語スコアが音節スコアに依存して生成されるようになったことの影響である。

ここで、ある単語認識率での未知語検出性能を調べるために、単語認識率が95%～99%のときのパラメータ μ_2 を、語彙数が10～1000までの複数の点で求めた。図5.3は、元の単語認識率が97%のシステム（パラメータ）での結果を示している。この図で“未知語入力”と“既知語入力”は、それぞれ未知語の入力に対する未知語検出率と、既知語（登録語）の入力に対する単語認識率を表している。未知語の検出率が上昇するようにしきい値を変えていくと単語認識率が減少していく様子が分かる。

図5.4は、未知語の検出率(correct rejection rate)と既知単語の棄却率(false rejection rate)の関係を示している。この結果から、まず $M = 1000$ で元の単語認識率が99%のシステムは、$M = 100$ で単語認識率が99% のシステムよりリジェクション（未知語の検出）の性能が良いことが分かる。しかし、語彙数が大きくなるほど、単語認識性能の違いによるリジェクションの性能への影響が小さくなっている。言い換えると、語彙数が十分大きいときは、単語認識率の少しの違いはリジェクションの性能にあまり影響しないといえる。図5.5は、未知語の検出洩れの割合(false acceptance rate)と、リジェクション実行時の既知単語の認識率（棄却されず正解認識である割合）を示している。未知語の検出洩れが特に少ない場合（例えば0~5%の間）について見ると、図5.4の場合と違って、元の単語認識性能の違いによるリジェクション実行時の既知単語

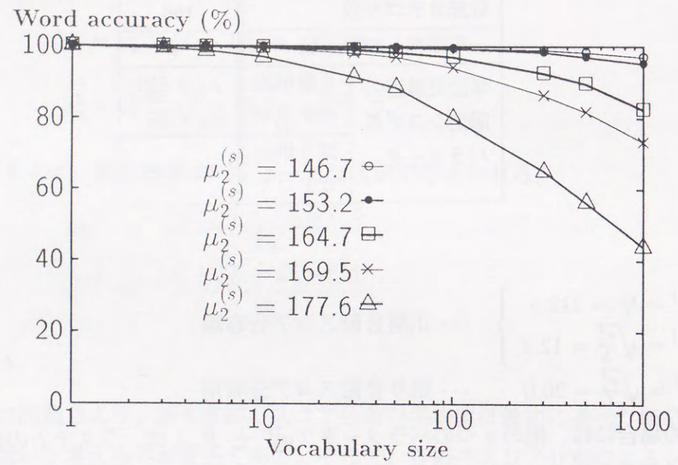


図 5.2: 語彙数と単語認識率の関係 ($\mu_1^{(s)} = 212.5,\ \sigma_1^{(s)} = 12.5,\ \sigma_2^{(s)} = 20$)

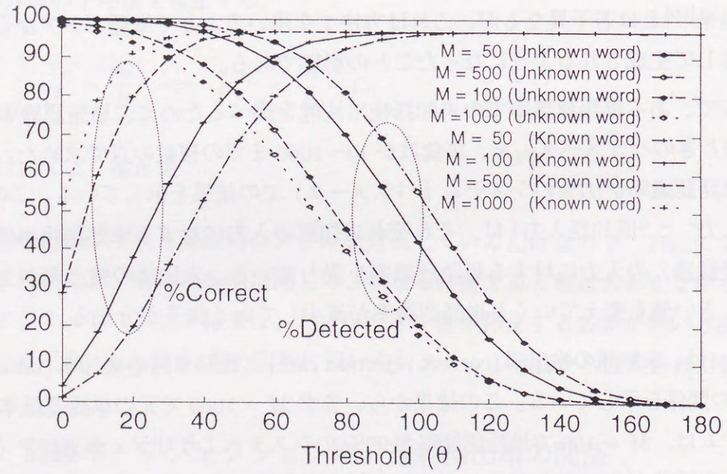
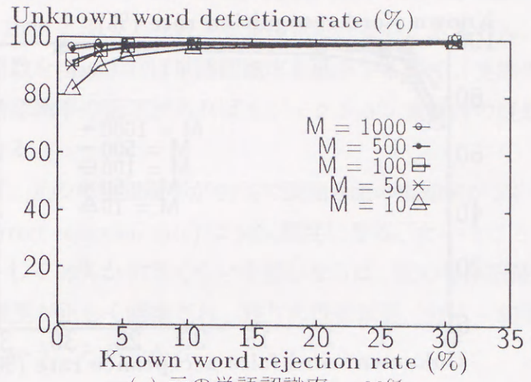
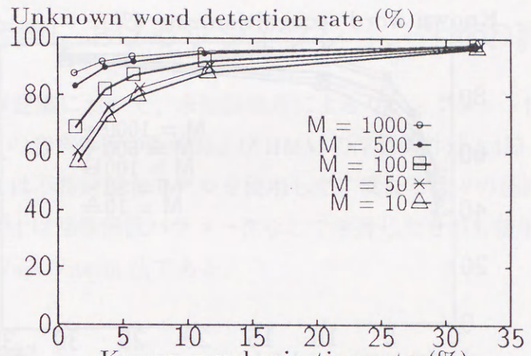


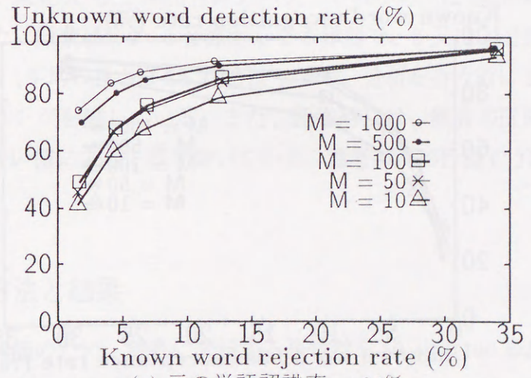
図 5.3: 単語認識率と未知語検出率
($\mu_1^{(s)} = 212.5,\ \sigma_1^{(s)} = 12.5,\ \sigma_2^{(s)} = 20$, 元の単語認識率 = 97.0%)



(a) 元の単語認識率 = 99%

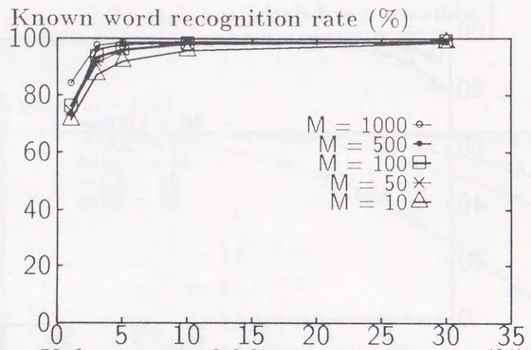


(b) 元の単語認識率 = 97%

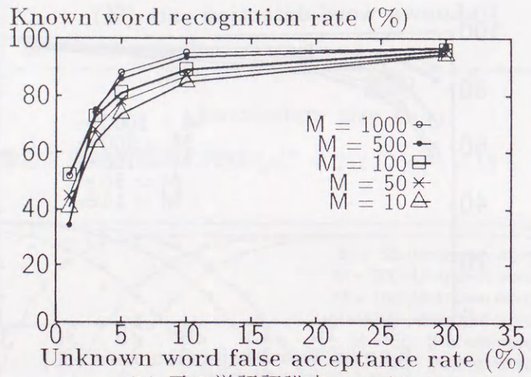


(c) 元の単語認識率 = 95%

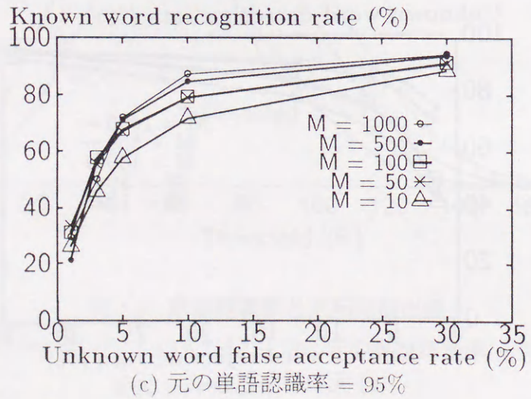
図 5.4: 未知語検出率



(a) 元の単語認識率 = 99%



(b) 元の単語認識率 = 97%



(c) 元の単語認識率 = 95%

図 5.5: 未知語処理時の単語認識率

の認識率の差は比較的大きい。また、その認識性能は、元の単語認識率が同じであれば語彙数が 10 ～ 1000 と違っても大きな差は見られない。もちろん、一般に同一の認識システムで単語数を増加すれば単語認識率も低下するので、実際のシステムで語彙数増加による単語認識率の低下があればリジェクション実行時の既知単語の認識率への影響は大きくなる。

大まかに言えば、元の単語認識率が 97%で既知単語の棄却率が 2%くらいのとき、未知語の検出率 (correct rejection rate) は 50%程度になる、ということができる。一方、未知語の検出率として 99%か 97%くらいを望むならば、元の単語認識率が 97%のとき、既知単語の 50%程度が正しく認識され、残りの既知単語（97% − 50% = 47%）は棄却されてしまう、ということが分かる。

5.3 単語音声におけるリジェクション性能の評価

孤立単語の音声認識において、未知語処理によるリジェクション性能の評価を行った。認識システムの音声分析の条件および HMM の学習条件は 4.4節と同じである。本実験では、HMM は不特定話者モデルを使用した。異なる種々の認識率に対する評価を得るため、HMM は回帰係数パラメータなしで学習したものも使用した。 認識アルゴリズムは One Pass Viterbi 法である。

5.3.1 音声資料

評価用として、東北大-松下単語音声データベース中の 212 単語集合の男性話者 15 名の発話を用いた。語彙は、2～8 音節からなる単語で、3 又は 4 音節の単語が全体の 85%を占めている（平均の長さは 3.6 音節)。音声は、分析を行う前に 24kHz から 12kHz にダウンサンプリングを施している。また、認識時には、無音の区間はあらかじめ定めたパワーのしきい値によって取り除いている。参考のため付録 C.2に単語認識実験の結果を示す。

5.3.2 実験方法と結果

未知語検出の評価のため、辞書に登録する語彙数を 50 又は 100 単語として、残りの単語を未知語とした。辞書に登録する単語は 212 単語からランダムに選択し、それぞれの語彙数に対して、単語集合が異なる複数セットの辞書を用意した。未知語モデルとしてのスコアは、連鎖制約のない連続音節認識によって求めた。実験では、式 (5.2) のしきい値は全ての話者に共通として、既知単語の棄却率 (false rejection rate) と未知

語の検出率 (correct rejection rate) の関係を調べた。

図5.6に示した結果は、(a) と (b) は語彙数が異なり、(a) と (c) は語彙数は同じで認識精度が異なっている。図の中の実線は、元の単語認識率を95%、97%、99%と仮定したときのシミュレーションの結果で、5.2.3節の方法1を用いている。ここで用いた P_{cc} の値は、システムの辞書から近似的に求めた（212単語の語彙に対して $P_{cc} \approx 0.05$）。この結果で、実音声による結果とシミュレーションの結果はほぼ同じような関係を示しているが、実音声による結果は語彙の選択の内容によってリジェクションの性能にややばらつきが見られる。また、この実験でも前のシミュレーションの結果と同様に、元の単語認識精度がリジェクションの性能に顕著に影響することが分かる。

5.4 文音声におけるリジェクション性能の評価

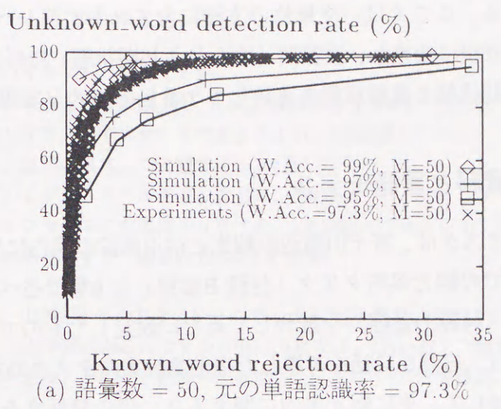
第4章で述べた未知語処理の方法は、入力音声の全ての区間が、文法で許される未知語や冗長語、または登録語の並びになっていると仮定する。しかしながら、文法外の発話や、あいまいな音声が入力されるような場合があり得るので、認識結果の信頼性が低いことを判定するための機能が必要となる。そこで、前述の単語発話のリジェクションと同様に、未知語処理の考え方を用いて文発話のリジェクションを行なうことを検討する。リジェクションの対象としては、主に文法外の発話や認識誤りになるような発話が考えられる。ここでは、そのような発話のリジェクション能力の実験的な評価結果を示す。

5.4.1 リジェクションの方法

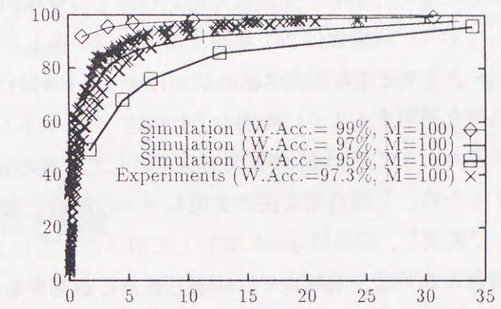
文発話の場合は、式(5.2)の $\boldsymbol{W}$ が文の単位を表すと考えてリジェクションの判定を行なえばよい。但し、リジェクションの対象として文法外の発話を考えた場合、式(5.2)の任意の音節系列 $\boldsymbol{s}$ の制約の代わりに、任意の単語（文節）系列を用いることも考えられ、未知語が含まれない発話に対してはより有効と考えられる[59]。そこで、式(5.2)の $\boldsymbol{s}$ の制約として次の2つを考える。

- 任意の音節連鎖
- 任意の音節連鎖と任意の文節連鎖の両方（任意の音節・文節連鎖）

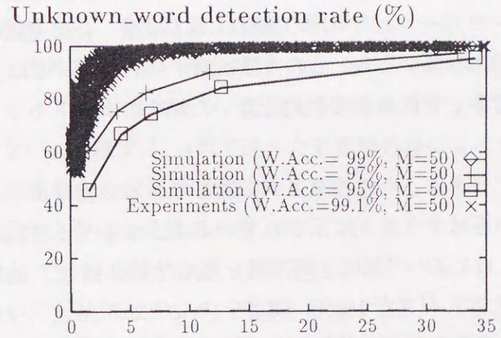
2番目の制約は、文節連鎖で対応できない音声区間のみ音節連鎖でカバーし、より高次の言語制約を用いることを目的としている。この場合、音節連鎖の制約は文節連鎖よりも弱いので、音節連鎖には若干のペナルティスコアを加え、文節連鎖を優先的な



(a) 語彙数 = 50, 元の単語認識率 = 97.3%



(b) 語彙数 = 100, 元の単語認識率 = 97.3%



(c) 語彙数 = 50, 元の単語認識率 = 99.1%

図 5.6: 孤立単語音声認識における未知語検出性能
(a),(b) : 10次元メルケプストラム係数
(c) : 10次元メルケプストラム係数 + 回帰係数

制約として用いる。ここでは、文単位の文法に含まれる文節レベルの制約を有限状態オートマトンの表現に変換し、文節間（すなわち初期状態）に任意の音節連鎖を許し、文節内文法の初期状態と最終状態を連結して2番目の制約を実現する。

5.4.2 音声資料と言語モデル

認識実験用のタスクは、富士山周辺の観光と宿泊施設の案内に関するもので、4章で述べた評価実験での観光案内タスク（付録B参照）と6章で述べる評価実験での宿泊施設案内タスク（付録D参照）を統合して新たに設定したものである。実験に用いたシステムの文法は、自然な発話を考慮した宿泊施設案内タスクの文法（6章で詳述。付録D.2参照）に対して、更に観光案内に関するユーザの発話文を受理できるように一部を変更し、拡張したものである。文法は文節レベルでの意味的な制約も含めて文脈自由文法で記述しており、語彙数は241単語である。言い直しや未知語に対処するため、間投詞のワードクラスと主な固有名詞のワードクラスに対して（全15種類）、4章で述べた未知語処理を適用するように特別な“未知語”のエントリを追加している。

本実験では、前述のように式(5.2)のsのモデルとして任意の音節連鎖または音節・文節連鎖を仮定するため、文脈自由文法の文節レベルの制約と音節連鎖の制約を有限状態オートマトンで表現し、認識処理時に並行して用いる。式(5.2)の音節連鎖の制約を用いた場合の照合スコアは、未知語や言い直しなどに対処するための未知語処理の結果として求まる[2]。有限状態オートマトンは全部で42状態（このうち2状態が、2音節以上の任意の音節連鎖を表す制約のためのもの）で、音節連鎖の制約を除いた有限状態オートマトンのアーク上の単語の総数は711単語、音節連鎖の制約に関するアーク上の総音節数が339個である。この有限状態オートマトンでは、評価文115文の中で101文を受理でき、音節連鎖の制約を除いた場合のテストセットパープレキシティは174である[3]。

評価用の音声データは、上述のタスクに関して2名の話者が115文づつ発話したものである。発話内容はテキストに示されているが、なるべく自然に話すようにして発声してもらった。115文の内訳は、間投詞を含んだ文が18文、助詞落ちを含んだ文が18文、言い直しを含んだ文が18文、倒置を含んだ文が16文、未知語を含んだ文が8文、倒置以外の文法外の文が8文あり（2種類以上の組合せが15文含まれる）、定型文は50文である。音声認識システムの辞書には、評価データに現れる間投詞のほとんど（14単語）を登録している。この評価データのうち文法で受理できる100文についての単語パープレキシティは75.7であった。図5.7に評価用文の例を示す（付録C.3参照）。

[2] 4.3.4節で述べた近似的な未知語処理アルゴリズムを適用。

[3] 音節連鎖の制約の音節を一単語と数えて含めて考えればパープレキシティは268となる。

文番号

2 富士急ハイランドって何ですか。
20 テニスのできるホテルは、えー、河口湖にありますか。（間投詞）
26 富士急ハイランドでスケートできますよね。（助詞落ち）
30 富士…、富岳の風穴って洞窟ですか。（言い直し）
39 ついていますか、ホテル西湖に食事は。（倒置）
49 ペンションマリエに食事はありますか。（未知語＝「ペンションマリエ」）
84 ホテルはありますか、温泉付きの。（文法外）

6 グラ…、山中ホテルの料金はいくらですか。（言い直し、未知語＝「山中ホテル」）
12 やま…、山中湖にあるんですか、鳴沢の氷穴は。（言い直し、倒置）
35 山中湖ホテル、そのー、泊ま…、宿泊したいんですが。（間投詞、助詞落ち、言い直し）
68 ペンションクレヨンにはペンション自慢の食事が付きますか。（間投詞、文法外）
86 いくらぐらいになりますか、河口湖の旅館は、えーと、宿泊の料金。（間投詞、助詞落ち、倒置）
107 温泉のあるホテ…、旅館に泊まりたいんですが。（言い直し、文法外）

図5.7: 評価用文の例

5.4.3 実験方法と結果

認識システムは4章で述べたSPOJUS-SYNO-Yを用いる。音声分析の条件およびHMMの学習条件は4.4節と同じで、HMMは話者適応化したモデルを用いる。

実験では、文法で受理できない文が多く現れるようにするため、文法において倒置を許す規則を全て削除した。その結果、評価文の中で文法で受理できる文は、倒置文、未知語を含む文、文法外の文を除いたもので、115文中で84文である。この84文に対する単語パープレキシティは70であり、倒置文を許すオリジナルの文法に比べても大きくは減少していない。実験では、4章で述べた未知語処理によって文中の未知語検出を行なうので、未知語を含んだ発話は文法外と考えないこととし、その結果、棄却対象となる文法外発話の文数は全発話の2割（23／115文）である。前述のように、式(5.2)のsの制約として音節連鎖か音節・文節連鎖を考えた場合と、比較のために定数を用いた場合についてもリジェクション性能を評価する。リジェクションの評価実験では、SPOJUS-SYNO-Yの認識結果として得られる累積対数尤度L_wと、未知語処理部によって求められる累積対数尤度L_uを、式(5.2)によるリジェクションの判定に用いる。これらの尤度は発話の長さに依存するが、閾値の設定を容易にするためフレーム長による正規化スコアによって判定を行なう。

図5.8は、文法外の発話のリジェクション性能を示しており、文法で受理できる文を誤って棄却した割合(false rejection rate)と、文法外の文の正しいリジェクションの割合

(correct rejection rate) との関係を示している。“Constant threshold” は定数、“Free-syllable likelihood ratio” は音節連鎖の尤度、“Free-syllable/phrase likelihood ratio” は音節・文節連鎖の尤度、をそれぞれ用いたリジェクション法に対応する。この図では、一般に曲線が上に凸であるほどリジェクションの性能が高く、(0,0) から (100,100) を結ぶ直線に近付くほど、リジェクションの判定基準がランダムに近く、リジェクションの性能が低いといえる。この結果から、単に定数を用いたリジェクション法は、他の方法に比べて顕著に性能が低いことが分かる。また、式 (5.2) の s の 2 種類の制約については、棄却誤り率が低いところでは音節・文節連鎖制約を用いた方が良いことが分かる。ところで、図 5.8の結果をみると、false rejection rate がある程度以上 (35%) のところで音節・文節連鎖による結果の折れ線のプロットの間隔が極端に離れているのが分かる。この理由は、音節・連鎖制約による方法では、約半数の発話が文法による認識結果と全く同じになっていて、閾値を 0 以上にしたときそのような発話が一度に棄却されるためである。

ここで、図 5.8の音節連鎖の尤度を用いたリジェクションの実験結果から、このシステムに対して音声入力を繰り返し行なったときの全体的なリジェクション性能について考える。実験結果では、false rejection rate($= p_f$) と correct rejection rate($= p_c$) の関係が常に $p_c > p_f$ (特に閾値を選べば $p_c \gg p_f$) となっているので、結果として棄却されなかった発話の中に棄却対象の文の発話が含まれる割合は、棄却前よりも減少することが分かる。例えば、使用した評価データの 115 文集合には棄却対象の文が 20% (23 文) 含まれるが、$p_f = 10\%$、$p_c = 90\%$の条件でリジェクションを行なうと、正しい棄却が約 21 文 (23×0.9)、誤った棄却が約 9 文 (92×0.1) となり、棄却対象の文が棄却されなかった発話に含まれる割合は約 2.4% (2/85 文) に減少する。図 5.9 は、音節連鎖の尤度を用いた方法でのリジェクションの閾値と認識性能の関係を示している。“Recognition rate” は、棄却されなかった発話に対する正解認識率を求めたものである。但し、正解認識とは、間投詞の挿入、脱落、置換や、助詞の誤りを無視し、未知語が認識結果に含まれるときはそれを無視したときに、それら以外の全ての単語が正しいような場合としている。“Total rejection rate” は、全発話に対する棄却率を示している。この図から、例えば文法外の発話を 9 割ほど棄却するときのしきい値は−0.8 で、その時、正解認識を誤って棄却する割合は 10 数%程度になることが分かる。この実験では、もともと倒置文の割合が少ないため Recognition rate の向上の効果は小さいが、文法外の発話が多い状況では効果が期待できる。

上述の文法外の発話のリジェクションの実験は、倒置文が主なリジェクションの対象であったため、正しい認識結果が期待できない発話の棄却として考えた場合、条件は比較的緩いといえる。図 5.10は、文法で受理できる文 (115 文中の 92 文) の発話に

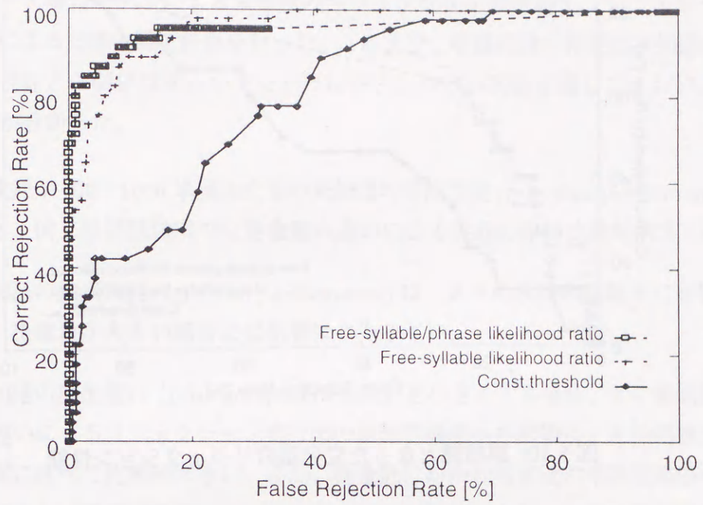


図 5.8: 文法外の文発話のリジェクション性能

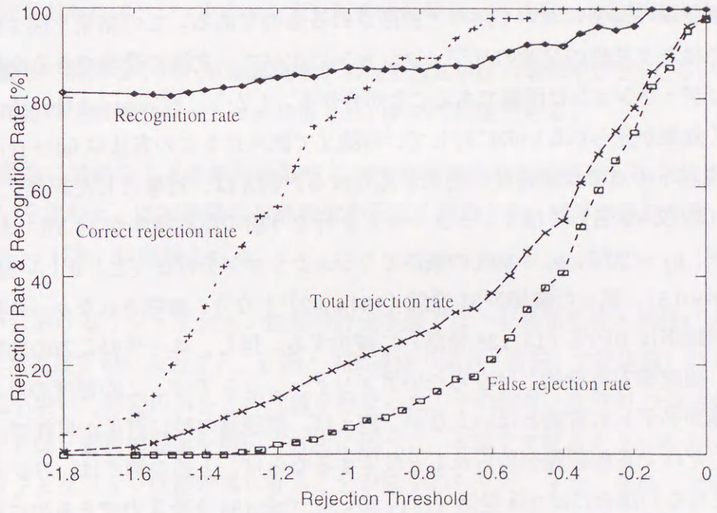


図 5.9: 閾値とリジェクション性能の関係

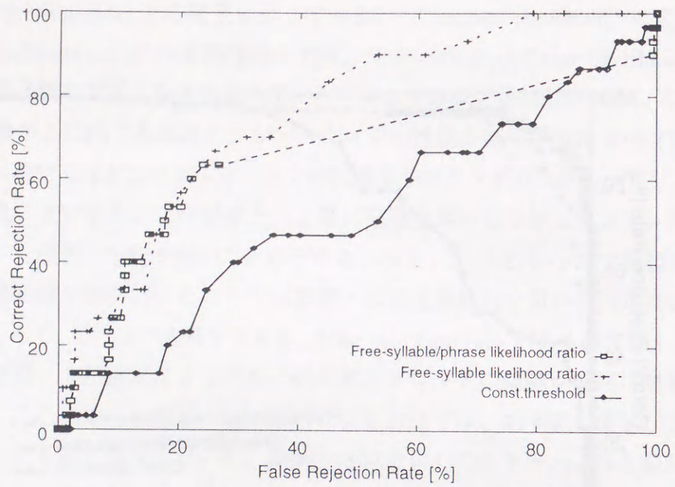


図 5.10: 誤認識となった文発話のリジェクション性能

対して誤認識となる発話を棄却対象としたリジェクション性能を示しており、正しい文を誤って棄却した割合 (false rejection rate) と、誤認識の結果の正しいリジェクションの割合 (correct rejection rate) との関係を示している。但し、誤認識の判定は、前述の正解認識の基準において誤りと判断されるものである。この結果を図 5.8 と比べると、やはり文法外の発話のリジェクションに比べて、文法で受理できる文発話の誤認識のリジェクションは困難であることが分かる。しかし、"Constant threshold" の方法が全く効果が見られないのに対して、本論文で試みた 2 つの方法は false rejection rate が 40%以下の点では同程度に効果が見られる。例えば、対象とした評価データの 184 発話（92 文×2 名）ではリジェクションを行なう前の誤認識率が 16.3%（30 発話）であるが、$p_f = 20\%$、$p_e = 50\%$の条件でリジェクションを行なうと、正しい棄却が 15 発話（30×0.5）、誤った棄却が 31 発話（154×0.2）となり、棄却されなかった発話に対する誤認識率は 10.9%（15/138 発話）に減少する。但し、ユーザの入力の効率を考えた場合、誤認識が致命的とはならないアプリケーションでは、この程度のリジェクション性能が必ずしも有効とはいえない。例えば、誤認識の時に訂正が可能で、一回の言い直しだけで正解認識が得られると仮定するならば、上述の例では、リジェクションを行なわない場合には 214 発話（154+30×2）で全 184 文を入力できるのに対して、リジェクションを行なう場合には 245 発話（123 + (30+31)×2）が必要になる。このことから、リジェクション性能の評価では一般にユーザの入力効率を考慮する必要があり、特に対話型のタスクでは、誤認識に対して効率的に訂正が行なえるような対話処理法について検討することも重要な課題といえる。

5.5 まとめ

サブワード単位モデルによる未知語のリジェクションの性能を、シミュレーションと実音声による認識実験で評価を行った。これまで、単語認識の精度と未知語のリジェクション性能との関係は明らかでなかったが、これらの実験を通して、おおよそ次のような事が分かった。

- 語彙数が 100～1000 単語あたりの未知語の検出性能 (rejection performance) をみると、同じ単語認識率でも語彙数の違いによる性能の差は比較的大きい。
- 未知語の検出性能 (rejection performance) は、元々の単語の認識率に影響され易く、語彙数が大きい場合には影響はやや小さい。
- 未知語の検出洩れ（false acceptance rate）を小さくする場合、元の単語認識性能の違いによるリジェクション実行時の単語認識率への影響は、未知語検出率への影響に比べて比較的大きい。また、語彙数にかかわらず元の単語認識率がほぼ同じであれば語彙数の違い (10 ～ 1000) によるリジェクション実行時の単語認識率における差は少ない。当然ながら、一般に同一認識システムで語彙数を増加させれば単語認識率も低下するので、実際のシステムで語彙数増加による単語認識率の低下があればリジェクション実行時の既知単語の認識率への影響は大きくなる。
- 語彙数に関係なく元の単語認識率が 97%で既知単語の棄却率が 2%くらいのとき、未知語の検出率 (correct rejection rate) は 50%程度である。
- 語彙数に関係なく元の単語認識率が 97%で未知語の検出率として 99%か 97%くらいを望むと、既知単語の 50%程度が正しく認識され、残りの既知単語（97% − 50% = 47%）は棄却される。

文音声におけるリジェクション性能の評価実験では、未知語処理と同様に音節連鎖を仮定した尤度を用いる方法と、音節と文節連鎖の両方（音節・文節連鎖）の尤度や定数の閾値を用いた判定の方法との比較を行なった。その結果、文法外の発話のリジェクションでは、定数の閾値を用いる方法に比べて、本論文で試みた 2 つの方法が顕著に良いリジェクション性能が得られることが確かめられた。また、音節・文節連鎖の尤度を用いる方法は、文法外の発話の誤った棄却をかなり少なく抑える場合には有利であることが分かった。

第6章

自然な発話のための照合・解析法の比較

6.1 はじめに

音声対話などにおける自然な発話には、4章で扱った間投詞や未知語の他に、言い直しや言い淀みなどの音声言語に特徴的な現象がある。また、従来の読み上げ音声を対象として作成された文法では、音声言語に現れやすい倒置や助詞落ちなどの現象を含む文を解析できないことが多い。そこで、自然な発話を扱うにはどのような音声認識手法が有効であるかを考えなければならない。

まず第一に、これまで扱っていなかった言い直しや言い淀みを処理する必要がある。その場合の問題は、間投詞や未知語と同様に言い直しや言い淀みの同定であるが、これまで主に採られている方法は、必要な単語のみを認識するワードスポッティング法の適用によって、言い直し等の区間に対する評価スコアを明示的に与えない方法である。一方、4章で述べたように未知語処理を適用することによって、それらの区間に相当する評価スコアを直接的に推定することが考えられる。前者の方法は特に言語処理での計算効率は良いといえるが、後者は認識スコアの正規化が必要ないためより最適な結果が得られると予想される。また、統合的なアルゴリズムではビームサーチ法による処理の削減が容易なため、後者の方法でも効率的な処理を行える可能性がある。本研究では、そのような観点から有効な方法について検討している。

これまでの音声認識システムは特に読み上げ文（朗読音声）を対象としており、少し型からはずれた非文法的な文は解析できない。そこで、より制約を緩めた文法を用いたり、文を解析できない場合に対処する方法などが必要になる。制約を緩めた文法を用いる場合は、解釈のバリエーションや曖昧さが増加し、その分解析処理の負担が増加する。一方、解析できない場合に対処する方法では、認識時のスコアが不十分なときに、受理する文の範囲を広げる弛緩法が提案されている[89, 90]。しかし、一般に認

識スコアは、その仮説による音声入力の尤度を求めているため、信頼性のスコアとして直接用いることは困難で、弛緩法を適用するための規準の設定は一般に難しい。まず、次節以降では、前者の制約を緩めた文法を用いた場合に、自然な発話に対して効率的で精度の良い処理を行うためのシステムの構成について検討する。

6.2 自然な発話のための照合・解析法

前述のように自然な発話には様々な特徴がある。そこで、主な音声・言語現象に対する音声認識システムの音声・言語処理のアプローチについて考える。

一般に言い直しや言い淀みなどについては、音声理解を目的としたアプリケーションでは間投詞と同様に不要語と考えることができる。不要語に対処するには、一つのアプローチとしてワードスポッティングベースの方式が考えられる。この場合、言語解析では必要な単語の仮説の集合（ワードラティス）だけを対象として処理することができる。しかし、ワードスポッティングでは確からしい単語の境界を推定する必要があるので、セグメンテーションの精度や候補を選択するためのスコアリングが重要である。一般には、そのような問題を改善するために、単語の前後に対して非キーワード（non-keyword）の音響的なモデル (filler model) を適用したり、簡単な言語モデルを仮定するなどして、発話全体に対して評価を行う方法が用いられる。本論文の 3.4.2節で検討したワードスポッティング法の改良は、基本的に後者の考えに基づくものである。

不要語を扱うための別のアプローチとしては、言い直しや言い淀みなどの音響／言語的な分析によってそれらを検出する試みがある[91, 69]。言い直しとポーズとの関係や言い直しの繰り返しパターンなどの特徴は、音声認識での適用が考えられる。しかし、それらの特徴だけでは十分な検出精度は得られていない。

より正確に言い直し等を処理するには、言語レベルの解析も含めた厳密な照合が必要と考えられる。そのための容易な解決法として、4章で述べた未知語処理法の利用が考えられる。未知語処理法を用いる場合の問題は、先見的な知識を入れても言い直しが現れる部分の予測は極めて困難と予想されることで、一般に文中のあらゆる部分での言い直しの出現を仮定する必要がある。しかし前述の未知語処理法で近似的な方法を用いると、このように未知語仮説が多く生成される可能性がある場合にも計算量の増加の問題はほとんどない。

言語処理では、これまで特に読み上げ文を対象として作成された文法に基づいて解析を行っているが、自然な発話に現れる非文法的な文も扱えなければならない。不完全な表現を扱うための頑健な解析を実現するには、意味的な制約に基づく文法を用いたり[84, 88]、解析できない文に対して部分的な解析を行う partial parsing のアプローチ[84, 85]を用いることが考えられる。従来の定型的な文の場合には構文主導のアプローチによって成功しているが[41]、意味的な言語制約に基づく場合には一般に構文的な制約が緩和されるため、必ずしも構文主導型の方法が良いとはいえない。そこで信頼性の高い単語や文節候補から先に解析を進める Island-Driven 法 (島駆動方式) の適用も考えられる。文頭から順に解析を進める Left-to-Right 法との比較では、定型文の文法で言語的な制約が同等である場合には、探索空間が大きければ両者の認識精度に差のないことが確かめられているが[92]、自然な発話を扱う場合については十分検討されていない。

以上の議論から、本章では、あらゆる倒置を認めた意味的な文法を用いて、認識システムにおける言語解析法や照合法の比較を試みる。比較するシステムの一方は、文節スポッティングに基づく方式である。この方式では、後述する Island-Driven 法と Left-to-Right 法による解析法の比較を試みる。もう一方のシステムは、3章で定型的な文に対して有効性が示された統合的な One Pass アルゴリズムに基づく方式である[49]。この方式と文節スポッティングに基づく Left-to-Right 解析法による方式との基本的な違いは、文節ラティスを介していないことである。One Pass アルゴリズムに基づく方式では、前述のように未知語処理の有効性も確かめる。一方、文節スポッティングに基づく方式でも未知語処理と同様な情報を加えた場合の有効性について検討する。これらの異なる方式による実験は全て等価な言語的制約を用いて行い、認識精度及び処理効率を比較する。

6.3 文節スポッティングに基づく連続音声認識システムの実現

6.3.1 文節スポッティング法

文節スポッティングではセグメンテーション及び抽出の精度が重要である。そこで前述のように発話全体の照合に基づく方法を適用するため、図 6.1のオートマトンに示されるような構文制約を考える。このオートマトンは任意の音節系列と文節の連結を表現しており、文節の開始状態にある音節のループはスポッティングする文節の前の区間の音声に対応した言語モデルを仮定している。文節スポッティングは、基本的に 2.4.4 節で示したオートマトンの制約に基づいた連続音声認識法によって実現する。文節候補は、処理するフレーム毎に、文節の終端から文節の先頭までのバックトレースを行なうことによって一つの候補ずつ求めることができる。一フレーム当たり一つの最適な候補だけでは検出漏れが起こり易いので、バックトレース時に、オートマトン

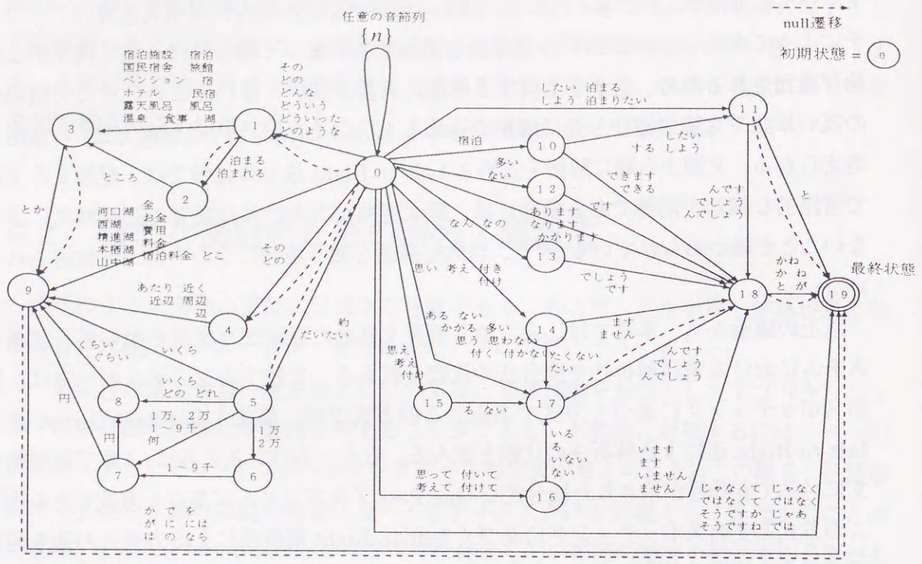


図 6.1: 文節スポッティングの認識のための構文

の各状態において単語毎に残されている累積スコアの上位 N個を全てたどっていくことによって、複数の候補を求める。

音声言語では特に助詞落ちを考慮しなくてはならないため、図 6.1のように助詞落ちを含めた文節文法を用いる。結果的に助詞や語尾変化の違いによるあいまいな文節候補が数多く生成され易いので、上述のような N-best 法はとくに重要になってくる。また、フレーム単位の処理では、隣接したフレームにおいて区間が似た同じ文節の候補が複数出てくることがあるので、隣接するフレーム内（実験では±64 msec、約 0.5 音節長）で同一の文節が検出されている時は、スコアが最も高い候補のみ残すようにする。このようにして、入力音声を文節ラティスに変換する。

6.3.2 Island-Driven 型解析法

不要語や言い直しを含んだ自然な発話を扱う場合、認識スコアが高い文節候補から解析を進めるボトムアップ的なアプローチが有効と考えられる。しかし、文節間の構文的な記述が複雑であれば island-driven 型の解析処理も複雑になる。そこで、格文法と同様な考えに基づいて、意味的な文節クラスの共起関係による簡単な文法記述を用いる。文法の構文、意味レベルの記述は、図 6.2の例に示すように 2 種類の規則からなる。(a) は語順（文節順）の制約を規定しない規則で、各々の規則が意味的にクラス分けさ


```
SENT :   VERB1   AT-LOC   ACCOM
SENT :   VERB1   ACCOM
SENT :   VERB2   OPTION   ACCOM
SENT :   VERB2   OPTION
SENT :   VERB3   HOWMCH   RATE
SENT :   VERB3   HOWMCH
```

(a) 文レベルの規則

```
VERB1    stay
VERB2    have
VERB3    costs
ACCOM    accom
ACCOM    accom     accom
AT-LOC   PLACE
AT-LOC   PLACE     near
PLACE    place
PLACE    place     place
OPTION   option
OPTION   option    option
HOWMCH   howmch
RATE     rate
RATE     ACCOM     rate
RATE     OPTION    rate
```

(b) 文節連接の制約のための文節間レベルの規則

(c) 文節クラスの例

意味的に等価な文節のクラス	代表的な文節の例（助詞は一部省略）
stay	宿泊したい, 泊まりたい
have	付く, 付いている
costs	（費用が）かかる, なる
accom	宿泊施設, ペンション
near	あたり, 周辺, 近辺
place	西湖, 山中湖, 河口湖
option	温泉, 食事
howmch	いくら, どれくらい
rate	料金, 費用, 宿泊料金

図 6.2: 文法の一部の例 （意味的に等価な文節のクラスを小文字で表す）

れた文を表現している。(a) のそれぞれの記号は意味的にまとめられた句を表し、実際は (b) で規定される文節列に対応する。(b) の規則は、文節間の修飾や並列構造（例えば、「宿泊の料金」「山中湖と河口湖」など）を表し、連接の制約を記述している。小文字の記号は意味的に等価な文節クラスを表し、(c) に示すような文節の集合に対応している。本章の実験で用いた文法を付録 D.1に示す。

実現された Island-Driven 法の、基本的な解析手順を以下に示す。

1. 動詞を含む文節候補を文節ラティスからスコアの高い順に最大で N 個（N: ビームサーチ幅）取り出し、新たな部分候補とする。

2. 部分候補が得られる毎に、つぎの操作を行なう。

   (2-a) 部分候補の解析結果に不完全な句が含まれるとき。

   句の内部の構文的な制約で接続可能な文節候補を、文節ラティスからスコアの高い順に最大で N 個取り出し、接続したものを新たな部分候補とする。

   (2-b) 部分候補の解析結果に未決定の句が含まれるとき。

   未決定の句の内部に含まれ得る文節候補を、文節ラティスからスコアの高い順に最大で N 個取り出し、接続したものを新たな部分候補とする。

   (2-c) 部分候補の解析結果が文形式になっているとき。

   認識結果の候補のリストに追加する。

図 6.3は、上記の解析手順を概念的に示したものである。上記の解析手順において、文の部分候補の展開は深さ優先で行なう。また、解析の途中では、文節間のスキップ区間（ギャップ）に対するスコアは仮定せず、区間の長さも制限していない。文法のあいまいさによって途中で部分的な候補が爆発的に増えることがあるので、部分候補のスコアを文節スコア（フレーム長で正規化されている）の和によって求め、スコア順で上位の候補のみ展開を行うよう制限を加えている。但し、部分候補はそれぞれ長さやスキップ区間の有無が異なり、スコアを直接比較することはできないので、部分候補に含まれる文節数毎に候補数を制限している。後述の実験では、全ての文節数について一定の制限候補数を設け、“制限候補数＝ビームサーチ幅 N” とした。深さ優先の探索なので、仮説の展開のたびに対応する文節数の仮説のスタックを調べ、同一の仮説がなければ登録して続けて展開を行なう。仮説を登録しようとしたときに仮説の数がビーム幅を越えた場合には、スコア順で最下位になるスタック内の仮説と比較し、登録しようとする仮説の方がスコアが高い場合にはスコアが低い方の登録された仮説と入れ換えて登録し、その後の仮説の展開を継続する。結局、この枝刈りの制限によって、一つの部分候補からのビームサーチ幅は実質的には更に制限されている。

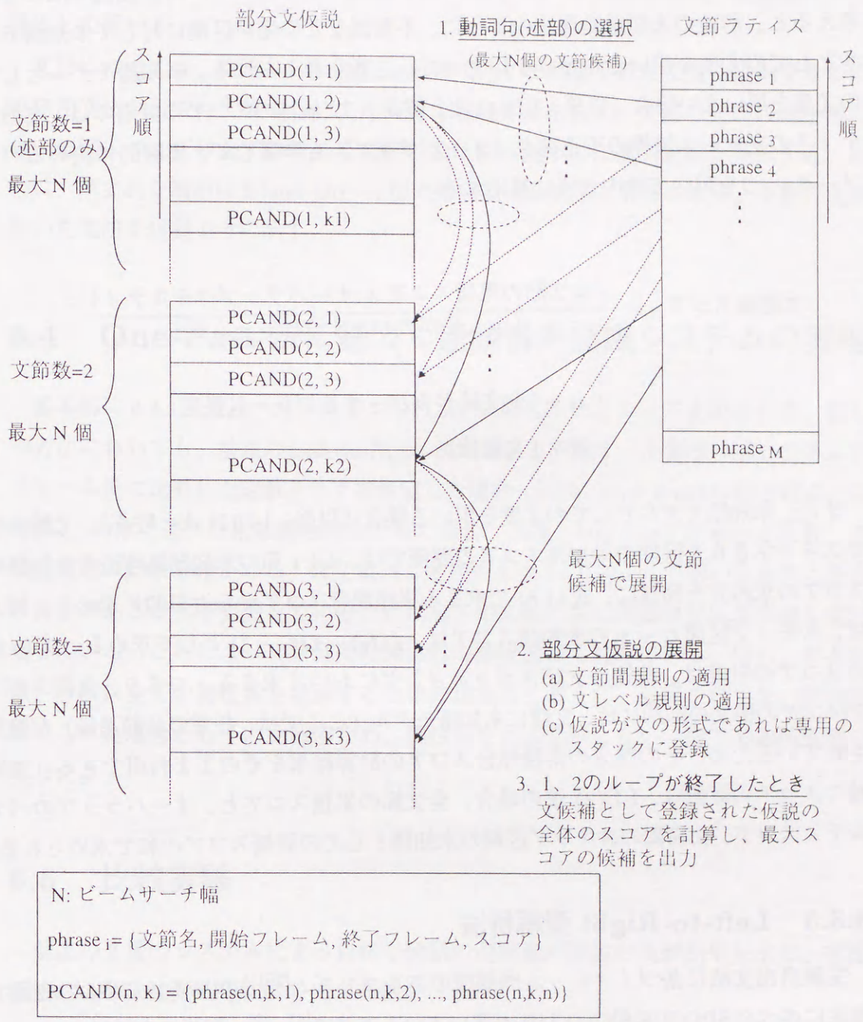


図 6.3: Island-Driven 型解析法の処理手順

最終的には複数の文の候補が求まるので、文を構成する各文節のスコアをもとに文の候補としてのスコアを評価する。もし発話に不要語が含まれていないと仮定すれば、発声区間長をほぼカバーしているような候補を優先すればよいが、不要語などが含まれる場合も考慮する必要がある。そこで、文候補としてのスコアの評価値の求め方を考えると、前述の未知語処理法のように、不要語などの発声区間に対して未知語モデルとしての尤度を用いるかどうかによって、二通り考えられる。未知語モデルとしての尤度を用いない場合（以後、I-D(1) 法と呼ぶ）、文候補のスコアの評価では、文節ラティスのフレーム境界の不正確さ（オーバラップ）も考慮して、実験的に決めるパラメータα、βを用いて次のように算出する。

$$\text{文認識スコア} = \frac{(\text{全文節の累積スコア} + \text{オーバラップペナルティ})}{(\text{全文節がカバーするフレーム長})} + \alpha \times (\text{全文節がカバーするフレーム長}) - \beta \times (\text{文節数})$$

また、未知語モデルとしての尤度を用いる場合（以後、I-D(2) 法と呼ぶ）、文節の間でスキップされた区間の未知語スコアの評価では、4.3.4 節の未知語処理法の未知語のスコアの求め方と同様に、式 (4.8) で求まる累積照合スコアから近似的に求める。例えば、スキップ区間 $t_1 \sim t_2$の未知語スコアは、$L(t_2) - L(t_1 - 1)$ として求める。式 (4.8) のスコアの計算は、前述の文節スポッティングにおいて求まる。つまり、文節スポッティングの構文の制約には、文頭に未知語モデル（ここでは、任意の音節連鎖）が適用されているため、その部分の累積照合スコアの計算結果をそのまま利用できる。文候補のスコアの評価は、I-D(2) 法の場合、全文節の累積スコアと、オーバラップのペナルティスコア、文節間のスキップ区間の未知語としての評価スコアの和で求められる。

6.3.3 Left-to-Right 型解析法

文脈自由文法に基づくフレーム同期型のアルゴリズム[26]を用いる。これは 3.2節で簡単に述べた SPOJUS-SYNO I/II が用いている方法で[41, 22]、解析の方法は Earley のアルゴリズムに基づく top-down 型の方法である。自然な発話を扱う場合、不要語の発話区間に対するスコアの推定が必要なことと倒置文への対処が必要な点が従来と異なる。不要語の発話区間としてのスコアの推定は、前述の Island-Driven 法の場合と同様に 2 種類考える。未知語モデルとしての尤度を用いない場合（以後、L-to-R(1) 法と呼ぶ）、不要語の発話区間としてのスコアは、一フレーム当たりの定数スコアをもとに近似する。但し、話者や発話内容による尤度の変動の影響を抑えるため、定数スコアは、文節スポッティング時に得られる音節連鎖による認識スコア（対数尤度）をフレーム長で正規化して求める。未知語モデルとしての尤度を用いる場合（以後、L-to-R(2) 法）は、前述の Island-Driven 法の場合と同じように、スキップする区間の未知語スコアを算出して用いる。

文法については、前述の Island-Driven 法で用いる倒置を許した文法規則をもとに文脈自由文法の形式に変換する。また倒置に対応するために、文節の並びの順序制約がない規則については全ての可能な文節の並びが可能な形に構文規則を展開する。結果的に、構文的な制約は Island-Driven 法で用いる文法規則と等価になる。本章の実験で用いた文法を付録 D.2に示す。

6.4 One-Pass 法に基づく連続音声認識システムの実現

基本的に 4.4.1節で述べたシステム SPOJUS-SYNO-Y によって実現される。但しこの方法においても、前述の Left-to-Right 法の場合と同様に、不要語の部分のスコアをフレーム長に比例した定数スコアで仮定した場合（以後、One-Pass(1) 法と呼ぶ）でも評価し、4章で述べた未知語処理を行った場合（以後、One-Pass(2) 法）と比較して未知語処理の効果を確かめる。構文規則は前述の Left-to-Right 法で用いるものと基本的に同じであるが、不要語を認めるために全ての文節の間に未知語が出現可能な規則になっている。これによって登録間投詞以外の間投詞にも対応できる。但し、本実験では、一般に全ての間投詞を登録するのは非現実的と考えて、登録されてない間投詞が用いられた場合としての比較のため、間投詞を辞書登録しないで全て未知語処理で対処した。

6.5 比較実験

前述の 3 種のシステムによる自然な発話の連続音声認識の実験結果を示し、言語解析法および未知語処理に関しての評価を行う。

6.5.1 評価タスク

評価用のタスクは、4.4節の未知後処理に関する実験で対象とした「富士山観光案内」のサブタスクで、「宿泊施設」の案内に関する話題に絞って評価を行なった。語彙数は約 120 語で、間投詞や未知語部分を無視した場合、前述の文法によるテストセットの

＜間投詞を含む文例＞
No.21 その周辺に、【えーと】、どのような民宿がありますか。
No.22 西湖で、【え】宿泊したいんですが。
No.24 【あの】、お金は、いくらかかると思いますか。
No.25 その周辺に、【えーと】、どのような民宿が【あーー】ありますか。

＜助詞落ちがある文例＞
No.53 その近くには、どんな旅館あるんですか。
No.54 どこが多い。泊まるところ。 (倒置)
No.55 つきますか。食事。 (倒置)
No.59 そのペンション【あーー】、いくらかかりますか。

＜言い直しがある文例＞
No.63 民宿に、(食‥、) 食事とかはついているんでしょうか。
No.64 温泉 (に、) は、ついているんでしょうか。
No.68 旅館は、(おか) 【あーー】宿泊料金がいくらぐらいになりますか。
No.69 (やまな、) 【えーと】西湖のどこにありますか。

図 6.4: 評価用質問文の例 ((倒置) は倒置を含む文)

単語パープレキシティは 41.1 である。なお、語彙数 500 の「富士山観光案内」のタスクの定型文のテストセットパープレキシティは 29.3 であった (4.4.1節参照)。

6.5.2 音声資料

認識に用いた音節 HMM と学習法は、4.4節の評価実験と同じで、HMM の話者適応化も同様に適応化用 20 文 (Fuji20) を用いて行っている。評価用の音声データは、前述のタスクに関して 2 名の話者が 70 文づつ発話したものである。発話内容はテキストに示されているが、なるべく自然に話すようにして発声してもらった。70 文の内訳は、(辞書登録されてない) 間投詞を含んだ文が 20 文、助詞落ちを含んだ文が 10 文、言い直しを含んだ文が 10 文、倒置を含んだ文が 9 文あり (2 種類以上の組合せが 13 文含まれる)、定型文は 36 文である。評価に用いた文の例を図 6.4に示す。また全発話文のリストを付録 D.3に示す。

6.5.3 実験結果

評価実験では、それぞれの認識方法の比較のために、ビーム幅、ペナルティスコア、スコアの重み係数などのパラメータを評価用データの実験結果から事後的に最適に決

表 6.1: 文節ラティスの質

特徴パラメータ	検出順位と文節検出率 (%) 1位	≤2	≤5	≤10	検出洩れ文節数	平均文節候補数
MEL	58.4	79.0	95.1	98.7	3 (1.4%)	2451
MEL+RGC	62.9	85.3	96.8	98.2	4 (1.8%)	2429

めた。

初めに、文節スポッティングに基づく認識法で得られた文節ラティスの質を調べた。前述の文節スポッティング時のペナルティ値は、実験的に一フレーム当たり−2.0 とした。文節ラティスの質は、文節単位のラベルをもとに、各文節境界内での正しい文節候補の検出順位で評価した。表 6.1に結果を示す。特徴パラメータの MEL は 10 次元のメルケプストラムを表し、RGC はその回帰係数を表している。評価時の文節境界としての許容誤差は最大約 130msec とし、各発話の文節境界は、音節系列の教師ありで話者適応した音節 HMM を用いて、Viterbi アルゴリズムによる最適パスをバックトレースすることによるセグメンテーションによって自動的に求めた。表より、ほとんどの文節が 10 位以内で検出されているのが分かる。上位には助詞だけが異なる候補が多かった。

Island-Driven 法の文認識スコアの係数パラメータ値を決定する予備実験では、用いる特徴パラメータ (MEL または MEL+RGC) 毎に最適な値を決定した。MEL+RGC の特徴パラメータを用いた場合の予備実験では、定義式の文節数に関する項 (係数βの項) を零としてフレーム長に関する項の係数αを可変させた場合には文理解率 (後述) は最大で 60.7%で、更にβを最適化することで 70.0%となった。係数パラメータ値の話者による影響は少なく、実験では話者共通で最適値を求めた。

表 6.2に認識結果をまとめた。この表で、文認識率は間投詞、言い直し以外が正確に認識されている割合を示し、文理解率は認識結果に対する文節クラスの系列が正しく、名詞などが意味的に正しく認識されている割合[1]を示している。

6.5.4 実験結果の考察

文理解率を比較すると、特徴パラメータとして MEL を用いた場合は、L-to-R(2) 法が若干良く、未知語モデルのスコアの使用も効果があった。一方、MEL+RGC を用いた場合は、未知語モデルのスコアを用いた方法が、用いない方法よりも全体に 10%前後高い性能が得られ、その効果が顕著に現れている。MEL+RGC を用いた場合の文理

[1]助詞の違いだけでは文節クラスの系列は変わらないので、助詞誤りも含まれる。

表 6.2: 認識性能の比較

特徴パラメータ	方法	文節認識率 (%)[1] %COR	文節認識率 (%)[1] %ACC	文認識率 (%)	文理解率 (%)
MEL	I-D(1)	55.4	53.6	22.9	65.7
	I-D(2)	60.9	54.5	23.6	69.3
	L-to-R(1)	67.4	56.6	27.9	69.3
	L-to-R(2)	63.8	59.0	25.0	74.3
	One-Pass(1)	67.9	58.1	28.6	61.4
	One-Pass(2)	63.1	59.0	23.6	67.1
MEL+RGC	I-D(1)	64.3	57.9	30.7	70.7
	I-D(2)	64.9	61.1	27.9	80.0
	L-to-R(1)	69.5	55.0	28.6	68.6
	L-to-R(2)	69.7	64.9	27.9	83.6
	One-Pass(1)	72.6	60.4	35.0	65.7
	One-Pass(2)	77.6	72.4	38.6	77.9

[1] %COR(正解率) = 正解文節数 / 発話文節数 × 100.0
%ACC(認識率) = (正解文節数 − 挿入文節数) / 発話文節数 × 100.0

I-D 法のスコアの重み係数： $\alpha = 0.02,\ \beta = 0.5$ (MEL)
$\alpha = 0.06,\ \beta = 1.15$ (MEL+RGC)

L-to-R 法のペナルティスコア： −4 / frame (MEL)
−12 / frame (MEL+RGC)

One-Pass(1) 法の定数スコア： −46 / frame (MEL)
−100 / frame (MEL+RGC)

One-Pass(2) 法のペナルティスコア： −20 / 音節 (MEL)
−40 / 音節 (MEL+RGC)

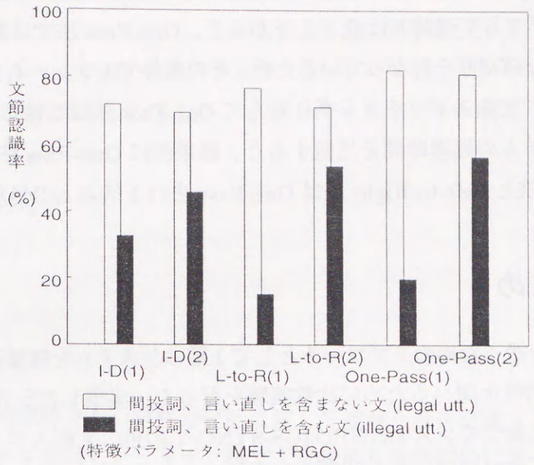


図 6.5: 各手法の認識性能の比較

解率で各方法を比較すると、上位 3 つは、未知語モデルのスコアを用いた方法である L-to-R(2) 法、I-D(2) 法、One-Pass(2) 法となっているが、顕著な差はない。結果的に、本実験で用いたタスクのように文理解としての性能が重要である場合には、未知語モデルのスコアを使用したそれぞれの方法の優劣は、主に計算の効率に関わってくるといえる。後述するように、実現したシステムの処理時間による比較では One-Pass(2) 法が最も効率が良かった。

文節認識率 (%ACC) や文認識率で比較すると、MEL+RGC の動的特徴パラメータを用いた場合に顕著な差があり、One-Pass(2) 法が最も高い。文節認識率の差の原因を認識結果から検証すると、主に、One-Pass(2) 法は他の方法に比べて助詞などの文節の語尾の認識がより正確であったことが挙げられる。しかし、文理解率で評価した場合は、前節で述べたように助詞の誤りを考慮しないので、その効果が目立っていないことが分かった。このことから、より忠実に発話を認識する必要があるディクテーションのようなタスクにおいては、特に One-Pass(2) 法が有利になることが分かる。図 6.5は、各認識方法の文節認識率を比較したもので、間投詞や言い直しを含む文と含まない文の、それぞれの発話の種類毎に示している。Left-to-Right 法や One-Pass 法では、未知後モデルのスコアの使用によって、間投詞や言い直しを含む文に対する文節の誤認識が半分近く減少している。また、間投詞や言い直しを含む文については、One-Pass(2) 法と L-to-R(2) 法が同程度で文節認識率が最も高いが、定型文ではやはり One-Pass(2) 法が有利であることが分かった。なお、表 6.2の結果は、Island-Driven 法ではビーム

幅= 200、他の方法は全てビーム幅= 40 としたが、Island-Driven 法以外ではビーム幅を 10 まで下げても文理解率は低下しなかった。One-Pass 法では累積スコアによるフレーム同期的な枝刈りを行なっているため、その条件で1フレーム当たりの照合単語数を比較すると、文節スポッティングに対して One-Pass 法は 2 倍以下に留まっていた。実現したシステムの処理時間を比較すると、結果的に One-Pass 法が最も効率が良く、Island-Driven 法と Left-to-Right 法は One-Pass 法の 2 倍以上の処理時間を要した。

6.6 まとめ

自然な発話を扱うためのアプローチとして3種のシステムを構築し、認識手法及び未知語処理の有効性を調べるために比較実験を行った。実現したシステムは、One Pass アルゴリズムに基づくシステム SPOJUS-SYNO-Y の他、文節スポッティング法に基づく island-driven 型と left-to-right 型の解析法によるシステムである。これらの方法に対して未知語処理または未知語処理と同様な情報を用いた場合と用いなかった場合についても比較評価した。

間投詞や倒置、言い直しを含むテキストを自然な発話に近い形で朗読した音声データで評価実験を行なった結果、One-Pass 法とその他の認識法においても、不要語を含む発話に対して未知語モデルのスコアを用いることの有効性が示された。特に、未知語処理を組み込んだ One-Pass 法は、定型文で高い認識精度を得ながら、不要語を含む発話についても他の認識法と同等以上の性能が得られ、処理時間も Island-Driven 法や Left-to-Right 法に比べて半分以下に抑えることができた。このことから、未知語処理の適用や文法の制約の緩和により、One-Pass 法が自然な発話への適用に対しても有効であることが示された。しかし、評価データの自然発話の傾向や制約などによる影響も考えられるので、今後更にユーザ毎の自然発話の傾向やタスクの違いなどによる影響について調べる必要がある。今後の課題としては、間投詞や言い直し、言い淀みなどの音響的性質や出現の傾向を利用した音声認識システムの頑健性の向上が考えられる。

第7章

結論

本論文では、効率的でかつ自然な発話に対して頑健な音声認識システムを構築するためのアプローチを検討した。まず、構文・意味的な言語制約に基づいて言語処理を音声処理と統合することによって、従来の一般的な階層型のシステムに対して認識精度および効率を改善するためのアルゴリズムを提案した。さらに、未知語や冗長な語を含む音響・言語的な制約を音声処理に組み込むことによって、自然な発話に対して効果的な音声照合を実現するための認識アルゴリズムを提案した。これらの方法では、信頼性の低い認識結果の棄却という側面での評価や、他の代表的な音声認識アルゴリズムとの比較実験を通して、自然な発話における有効性を確認した。

構文・意味的な言語制約を記述するための言語モデルとしては、自然言語に対する記述が容易で効率的な解析法が知られる文脈自由文法の利用が一般的に用いられる。そこで、3章では、単語を照合単位とする音声処理部分と文レベルの処理である文脈自由文法の構文解析法を統合する方法の実現について述べた。その実現方法として2通り提案し、一つは拡張連続 DP 法の原理によって、単語毎の独立した音声照合を行なうワードスポッティング法に基づいた統合化を実現した。この方法は、従来の階層型のシステムと同様に語彙数のオーダの処理量を実現するが、統合化に伴ってワードスポッティング法に対しても文認識レベルのスコアを近似的に組み入れることが可能になった。そのため、評価実験では、従来の階層型のシステムに比べて文認識誤りが約 50%減少したうえ、処理時間も数分の一程度に削減される効果がみられた。もう一つは One Pass アルゴリズムに基づいた方法で、言語モデルと音響モデルに関してより最適な照合・探索を行うアルゴリズムによって実現した。2つの方法ともフレーム同期型のアルゴリズムで、ビームサーチ法や枝刈りの適用が容易であり、特に後者の One Pass 法では処理量の大きな節減の効果が得られた。しかし計算量の比較では、パープレキシティが約 10 のタスクの認識において、前者の方法は計算量が後者の半分くらいになり、文認識率は同程度の約 90%が得られた。また、HMM の継続時間分布を用いない場合は、

前者のワードスポッティングに基づく方法では約2倍、後者のOne Pass法では約3倍近く高速化できたが、この場合は前者の文認識率の低下が顕著で、後者の方法に比べて文認識率が4%程低くなった。このことから、音響的なモデルの精度が高い場合には前者の効率的な認識法が有効で、そうでない場合はより最適な探索を行う後者を用いた方が良いといえる。

4章では、上述の構文・意味的な言語制約に加え、未知語や冗長な語の言語的な制約を音声処理に統合した、効率的な未知語処理の実現及びその有効性の評価について述べた。未知語処理は、未知語を任意の音節の並びと仮定することによって、サブワード単位の音響的モデルを利用でき、サブワード単位の連続音声認識による未知語照合の処理をOne Pass法に統合することによって実現した。また、連続音声認識に未知語処理を組み込むために、厳密な方法以外に、拡張連続DP法と同様な原理に基づいて処理量を大きく削減する近似的な方法を提案し、評価実験によって近似による性能の差がほとんどないことを確認した。また、未知語処理の実験では、未知語の仮説に関する単純な制約を課すことによって検出精度の向上と過剰検出の低減に効果があることが分かった。

間投詞は、種類が豊富で、その他の一般の単語内の音節に比べて発音のゆらぎがあることが予想されるため、未知語処理による扱いについても検討した。評価実験では、対話中によく現れる間投詞のみを辞書登録する場合と、未知語処理により間投詞の検出を行う場合、それらを併用した場合について認識実験を行なった。その結果、文頭の接続詞などとの間で誤認識や挿入などが多いことが分かったが、文頭について間投詞が認識されるように辞書登録した場合、間投詞を辞書登録だけで対象した場合と同程度の文認識率が得られ、認識処理時間を節約できる効果があることを確認した。

5章では、4章で述べた未知語処理と同様な原理でサブワードモデルに基づいて発話のリジェクションを行なう方法の評価を行なった。まずシミュレーションによって、語彙数または単語認識率をパラメータとしてリジェクション性能との関係を求めた。一般的な傾向として、(1) 未知語の検出性能 (rejection performance) は、単語認識率が高いときには同じ単語認識率であれば語彙数の違い（10~1000）による性能の差が小さいのに対して、語彙数に関係なく元の単語の認識率の違い（95%~99%）による影響が大きい、(2) 未知語の検出洩れ（false acceptance rate）と単語認識率との関係は、あまり語彙数に依存せず元の単語認識率の影響が大きい、ということが示された。実音声による孤立単語音声の認識およびリジェクションの性能の評価も行なったが、その結果シミュレーションと同様な傾向が見られ、単語認識精度がリジェクション性能に顕著に影響することが確認された。また、文音声におけるリジェクション性能の評価実験を、未知語処理と同様に音節連鎖の尤度を用いる方法と、音節と文節連鎖の両方の尤度や定数の閾値を用いた判定の方法とで、比較を行なった。その結果、文法外の発話のリジェクションでは、定数の閾値を用いる方法に比べて、未知語処理と同様な方法が顕著に良いリジェクション性能が得られることが確かめられた。また、音節・文節連鎖の尤度を用いる方法は、文法外の発話の誤った棄却をかなり少なく抑える場合には有利であることが分かった。

6章では、3, 4章で提案したOne Pass法に基づく統合的なアルゴリズムと、その他の代表的なアプローチの全3種類のシステムにより、自然な発話を扱うために有効なシステム構成の検討を行なった。言い直しなどの不要語を処理するために一般に用いられるスポッティング方式に基づく方法では、文節スポッティングを、Left-to-Right型解析法またはIsland-Driven型解析法と組合せてシステムを実現した。残りの一つのシステムは、4章でも用いたLeft-to-Right型解析法に基づくOne Pass法である。これらに対して、未知語処理または未知語処理と同等な情報の利用の有無によって認識性能の違いを比較した。疑似的な自然発話データで評価実験を行なった結果、One-Pass法とその他の認識法においても、不要語を含む発話に対して未知語モデルのスコアを用いることの有効性が示された。特に、不要語に対処するために未知語処理を組み込んだOne-Pass法は、定型文で高い認識精度を得ながら、不要語を含む発話についても他の認識法と同等以上の性能が得られた。処理時間を比較するとOne Pass法はその他のシステムに比べて優位であり、未知語処理や文法の制約の緩和によって自然な発話に適用する場合にも有効な方法といえる。

今後は、音声対話を目的としてより柔軟な音声認識システムを考える必要があり、以下のような課題があげられる。

まず、音声認識のための言語処理において、文理解、文脈処理などの対話処理が用いる知識を共有して利用することである。現在は、文理解において用いる意味的な知識による制約と音声認識における言語的な制約が一致しないため、正しい解釈が得られないような認識結果を出力することがあり得る。

次に、不要語の処理において、より正確に不要語を検出するための特徴の分析とその情報の併用である。一般にこれらの語は韻律的にも特徴をもっているので、そのような情報の併用によって更に正確な検出が行なうことができるものと考える。また、間投詞のような語は、発話者の情報処理状態の指標を表すことが指摘されており[74]、発話中において全く情報が無いものではない。従って、意味解釈においての利用も考えるべきであろうし、このような語によってある程度発話される文型の推定も可能になり、音声認識に役立つものと考える。

また、未知語や信頼性の低い認識結果をリジェクトするためにより有効な、認識仮説に対する信頼性の尺度 (confidence measure) の推定法の検討である。現在の方法は、音

響的なモデルの尤度だけで未知語としての仮説のスコアを求めるため、発音のスペクトルのゆらぎやサブワードモデルの精度に依存して未知語仮説のスコアも変動し、十分な精度が期待できない。十分な信頼性を持った未知語処理を実現するには、より多くの知識源を考慮して仮説の信頼性の尺度を求めるような方法が必要と考える[77]。

謝辞

本博士論文に関する研究の全過程を通じて、大変有益な御指導と御助言を頂いた豊橋技術科学大学情報工学系の中川聖一教授に深く感謝の意を表します。

本論文をまとめるにあたり有益な御助言を頂きました豊橋技術科学大学情報工学系臼井支朗教授、井上克巳助教授に対し、厚く御礼申し上げます。

豊橋技術科学大学情報工学系中川研究室の方々には、日頃から御協力頂いた。特に、間宮康之君、伊藤敏彦君には音声認識実験で用いた文法や発話文の作成に協力して頂いた。ここに記して、感謝の意を表します。

音響モデルを構築するのに用いた音声資料は、ATR で収集された音声データベースと、日本音響学会の研究用連続音声データベースの一部であり、関係各位のご尽力に感謝致します。

参考文献

[1] D. H. Klatt: "Review of the ARPA speech understanding projects," *J. Acoust. Soc. Am.*, Vol.62, No.6, pp.1345-1366 (1977).

[2] 鹿野清宏, 樽松 明：音声理解研究の動向, 日本音響学会誌, Vol.42, No.12 (1986).

[3] J. Peckham: "Speech understanding and dialogue over the telephone: an overview of the ESPRIT SUNDIAL project," *Proc. of the DARPA Speech and Natural Language Workshop*, pp.14–27 (1991).

[4] 新田恒雄：ESCA-NATO 音声技術応用ワークショップ, 信学会, 第 2 種研究会, SPREC-93-3, pp.9–15 (1994.2).

[5] 中川聖一, 鹿野清宏：カーネギー・メロン大学における音声認識・理解研究の現状, 日本音響学会誌, Vol.42, No.9, pp.743–747 (1986).

[6] 岡田美智男：音声言語システムの研究動向と今後の課題, 日本音響学会誌, Vol.48, No.1, pp.33–38 (1992).

[7] 甘利俊一監修, 中川聖一, 鹿野清宏, 東倉洋一 共著：音声・聴覚と神経回路網モデル, オーム社 (1990).

[8] M. Tomita: "Efficient Parsing for Natural Language," Kluewer Academic Publishers (1986).

[9] 新美康永：音声認識, 共立出版 (1979).

[10] 古井貞熙：ディジタル音声処理, 東海大学出版会 (1985).

[11] W.A. Woods: "Optimal search strategies for speech understanding control," *Artificial Intelligence*, Vol.18, pp.295–326 (1982).

[12] T. Sakai and S. Nakagawa: "A speech understanding system of simple Japanese sentences in a task domain," 信学論, Vol.60-E, No.1, pp.13–20 (1977).

[13] U.R. Lesser and L.D. Erman: "A retrospective view of the HEARSAY-II architecture," *Proc. 5-th International Joint Conference on Artificial Intelligence*, pp.790–800 (1977).

[14] B.T. Lowerre: "The HARPY speech recognition system," *Ph.D thesis*, Department of Computer Science, Carnegie Mellon Univ. (1976).

[15] D.B. Paul: "Algorithms for an optimal A* search and lenearizing the searh in the stack decoder," *Proc. ICASSP*, pp.693–696 (1991).

[16] F.K. Soong and E.F. Huang: "A tree-trellis based fast search for finding the N best sentence hypotheses in continuous speech recognition," *Proc. ICASSP*, pp.705–708 (1991).

[17] 河原達也, 松本真治, 堂下修司：単語対制約をヒューリスティックとするA*探索に基づく会話音声認識, 信学論, Vol.J77-D-II, No.1, pp.1–8 (1994).

[18] R. Schwartz and Y.L. Chow: "The N-best algorithm: An efficient and exact procedure for finding the N most likely sentence hypotheses," *Proc. ICASSP*, pp.81–84 (1990).

[19] R. Schwartz and S. Austin: "A comparison of several approximate algorithms for finding multiple (N-BEST) sentence hypotheses," *Proc. ICASSP*, pp.701–704 (1991).

[20] I.S.Bridle, el al.："An algorithm for connected word recognition," *Proc. ICASSP*, pp.899–902 (1982).

[21] H.Ney："The use of a one-stage dynamic programming algorithm for connected word recognition," *IEEE Trans. Acoust., Speech & Signal Process.*, Vol.32, No.2, pp.263–271 (1984).

[22] 中川聖一，大黒慶久，橋本泰秀：構文解析駆動型日本語連続音声認識システム -SPOJUS-SYNO-, 信学論, Vol.72-DII, No.8, pp.1276–1283 (1989).

[23] A. Kai, S. Nakagawa, "A frame-synchronous continuous speech recognition algorithm using a top-down parsing of context-free grammar," *Proc. ICSLP 92*, pp.257–260 (1992).

[24] 川端, 鹿野, 北：音韻パープレキシティの提案, 日本音響学会講論集, 3-6-12 (1989.3).

[25] 村瀬, 中川：単語の共起確率を用いた音声認識と文脈自由文法を用いた音声認識の比較, 信学技報, SP90-25 (1990.6).

[26] 中川聖一：文脈自由文法のフレーム同期型構文解析法による連続音声認識, 信学論, Vol.70-D, No.5, pp.907–916 (1987).

[27] 中川聖一：拡張連続DP法による連続音声認識アルゴリズム, 信学論, Vol.67-D, No.10, pp.1242–1249 (1984).

[28] 北　研二, 川端　豪, 斉藤博昭：HMM音韻認識と拡張LR構文解析法を用いた連続音声認識, 情報処理学会論文誌, Vol.31, No.3, pp.472–479 (1990).

[29] K.Kita, W.H.Ward："Incorporating LR parsing into SPHINX," *Proc. ICASSP*, pp.269–272 (1991).

[30] M.Okada："An efficient One-Pass search algorithm for parsing spoken language," *IEICE Trans.*, Vol.E75-A, No.7, pp.944–953 (1992).

[31] W. Ward and S. Young: "Flexible use of semantic constraints in speech recognition," *Proc. ICASSP*, pp.II-49–50 (1993).

[32] 平田好充，中川聖一：連続DPおよびO(n)DP法によるHMMワードスポッティングの比較, 日本音響学会講論集, 2-8-7 (1990.9).

[33] 河原達也, 宗続敏彦, 三木清一, 堂下修司：会話音声中の単語スポッティングのための言語モデルの検討, 信学技報, SP94-28 (1994.6).

[34] 渡辺隆夫, 吉田和永, 畑崎香一朗：バンドルサーチ法を用いた連続音声認識の高速化, 信学論, Vol.J75-D-II, No.11, pp.1761–1769 (1992).

[35] 沢井秀文，米山正秀，中川聖一：大語彙単語音声認識の高速化のための種々の検討, 日本音響学会誌, Vol.43, No.11，pp.858–867 (1987).

[36] 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, Vol.J71-D, No.9, pp.1650–1659 (1988).

[37] 中川聖一，梅崎太造：O(n)DP法による連続数字音声の認識, 信学論, Vol.J66-D, No.11, pp.1318–1325 (1983).

[38] 中川聖一，平田好充：連続出力分布型 HMM の話者適応化による日本語音韻・音節認識，日本音響学会誌，Vol.47, No.7, pp.459–467 (1991).

[39] 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).

[40] 福村晃夫，稲垣康善：オートマトン・形式言語理論と計算論（岩波講座 情報科学 6），岩波書店 (1982).

[41] S. Nakagawa, Y. Hirata, I. Murase, T. Tanoue, "Comparison of syntax-oriented spoken Japanese understanding system with semantic-oriented system," *IEICE Trans.*, Vol.E 74, No.7, pp.1854–1862 (1991).

[42] S.Nakagawa, I.Murase, "Comparison of language models by context-free grammar, bigram and quasi/simplified-trigram," *IEICE Trans.*, Vol.E 74, No.7, pp.1897–1905 (1991).

[43] S. Nakagawa, "Speaker-independent continuous-speech recognition by phoneme-based word spotting and time-synchronous context-free parsing" *Computer Speech and Language*, Vol.3, No.3, pp.277–299 (1989).

[44] H. Ney, "Dynamic programming parsing for context free grammars in continuous speech recognition" *IEEE Trans.*, Vol.SP-39, No.2, pp.336–340 (1990).

[45] M. Okada, "A One-Pass search algorithm for continuous speech recognition directed by context-free phrase structure grammar" *Proc. ICSLP 90*, Vol.2, pp.1229–1232 (1990).

[46] 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，信学技報，SP92-108 (1992.12).

[47] V. Zue, J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff, "Integration of speech recognition and natural language processing in the MIT Voyager system," *Proc. ICASSP*, pp.I-25–28 (1991).

[48] 中川 聖一，甲斐 充彦: ワードスポッティング法を用いた文脈自由文法制御フレーム同期型 HMM 連続音声認識法，信学論，Vol.J76-D-II, No.7, pp.1329–1336 (1993).

[49] 中川 聖一，甲斐 充彦: 文脈自由文法制御による One Pass 型 HMM 連続音声認識法，信学論，Vol.J76-D-II, No.7, pp.1337–1345 (1993).

[50] Seiichi Nakagawa and Atsuhiko Kai: "A context-free grammar-driven, One-Pass HMM-based continuous speech recognition method", *Systems and Computers in Japan*, Vol.25, No.4, pp.92–102 (1994).

[51] 森屋，阿部野，山本，中川：対話予測を利用した音声による観光案内対話システム，信学技報，SP92-121 (1993.1).

[52] Lawrence Rabiner and Biing-Hwang Juang: "Fundamentals of speech recognition, " Prentice Hall, New Jersey (1993).

[53] L.R. Rabiner, J.G. Wilpon, and B.H. Juang: "A model-based connected digit recognition system using either hidden Markov models or templates," *Computer Speech and Language*, Vol.1, No.2, pp.167–197 (1986).

[54] C.H. Lee, L.R. Rabiner, R. Pieraccini, and J.G. Wilpon: "Acoustic modeling for large vocabulary speech recognition," *Computer Speech and Language*, No.4, pp.127–165 (1990).

[55] Wilpon J. G., Rabiner L. R., Lee C-H. and Goldman E. R.: "Automatic recognition of keywords in unconstrained speech using hidden Markov models", *IEEE Trans. Acoust., Speech & Signal Process.*, Vol.38, No.11, pp.1870–1878 (1990).

[56] Asadi A., Schwartz R. and Makhoul J.: "Automatic detection of new words in a large vocabulary continuous speech recognition system", *Proc. ICASSP*, pp.125–128 (1990).

[57] Richard C. Rose and Douglas B. Paul: "A hidden Markov model based keyword recognition system," *Proc. ICASSP*, pp.129–132 (1990).

[58] Bourlard H., D'hoore B. and Boite J-M.: "Optimizing recognition and rejection performance in wordspotting systems" *Proc. ICASSP*, pp.I-373–I-376 (1994).

[59] 河原達也，北岡教英，堂下修司：フレーズスポッティングに基づく頑健な音声理解，情報処理学会研究報告，94-SLP-4-6 (1994).

[60] F. Jelinek: "Continuous speech recognition by statistical methods," *Proc. IEEE*, Vol.64, pp.532–536 (1976).

[61] Y. Ariki and T. Kawamura: "Simultaneous spotting of phonemes and words in continuous speech," *Proc. ICSLP 94*, pp.2191–2194 (1994).

[62] 新美康永, 高橋一城, 小林 豊：音声認識結果の認識誤り区間と未知語区間の推定, 信学技報, SP95-31 (1995.6).

[63] 北, 江原, 森元：連続音声認識における未知語処理, 日本音響学会講論集, 3-5-3 (1991.3).

[64] 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, 信学技報, SP91-96 (1991.12).

[65] 井ノ上直己, 武田一哉, 山本誠一: ガーベジ HMM を用いた自由発話文中の不要語処理手法, 信学論, Vol.J77-A, No.2, pp.215–222 (1994-2).

[66] 渡辺隆夫, 塚田　聡：音節認識を用いたゆう度補正による未知発話のリジェクション, 信学論, Vol.J75-D-II, No.12, pp.2002–2009 (1992).

[67] 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, 信学技報, SP91-100 (1991).

[68] 花沢利行, 阿部芳春, 中島邦男：発話様式変動を考慮した連続音声認識方式の検討, 日本音響学会講論集, 2-7-13 (1994.3).

[69] 中川聖一, 小林　聡：自然な音声対話における間投詞・ポーズ・言い直しの出現パターンと音響的性質, 日本音響学会誌, Vol.51, No.3, pp.202–210 (1995).

[70] 小林 聡, 甲斐充彦, 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, 信学会, 第 2 種研究会, SPREC-92-3, pp.21–25 (1993.2).

[71] 黒岩眞吾, 武田一哉, 井ノ上直己, 山本誠一：機械との対話における発話分析, 信学技報, SP94-30 (1994.6).

[72] 上條俊一, 秋葉友良, 伊藤克亘, 田中穂積：音声対話データの分析と発話理解への応用, 情報処理学会研究報告, 94-SLP-3, pp.31–36 (1994.10).

[73] 市川 熹, 佐藤伸二：対話理解に対する抑揚情報の役割, 情報処理学会研究報告, 94-SLP-2, pp.51–58 (1994.7).

[74] 田窪行則：談話管理標識の機能と心的領域について, 信学会, 第 2 種研究会, SPREC-92-1, pp.57–66 (1992.7).

[75] 中川聖一, 越川 忠：最大事後確率推定法を用いた連続出力分布型 HMM の適応化, 日本音響学会誌, Vol.49, No.10, pp.721–728 (1993).

[76] 山田雅章, 伊藤史朗, 酒井佳一, 小森康弘, 大洞恭則, 藤田　稔：音声対話 CD-ROM 情報検索システム, 信学技報, Vol.93, SP93-21 (1993.6).

[77] Sheryl R. Young and Wayne H. Ward: "The role of higher-level semantic, pragmatic and discourse knowledge in recognizing and understanding new spoken words and phrases," *Proc. ESCA Workshop on Spoken Dialogue Systems*, pp.29–32 (1995).

[78] 速水 悟, 伊藤克亘, 田中和世：未知語処理のための言語的制約について, 日本音響学会講論集, 2-4-16 (1993.3).

[79] Hayamizu, S., Itou, K. and Tanaka, H.: "Detection of unknown words in large vocabulary speech recognition," *J. Acoust. Soc. Japan*, Vol.16, No.3, pp.165–171 (1995).

[80] 中川聖一, 大黒慶久, 村瀬 功：連続音声認識システムの評価法 —タスクの複雑性と文認識率との関係—, 信学論, Vol.J73-D-II, No.5, pp.683–693 (1990).

[81] 中川聖一：音韻認識率と単語認識率との関係, 情報処理学会論文誌, Vol.22, No.5, pp.488–496 (1981).

[82] S. Nakagawa and I. Murase, "Relationship among phoneme/word recognition rate, perplexity and sentence recognition and comparison of language models," *Proc. ICASSP*, pp.I-589–592 (1992).

[83] 森屋 裕治：日本語音声による観光案内対話システムの構築に関する研究, 豊橋技術科学大学 大学院 修士学位論文 (1993.2).

[84] W. Ward: "Understanding spontaneous speech: The phoenix system," *Proc. ICASSP*, pp.365–367 (1991).

[85] Stephanie Seneff: "Robust parsing for spoken language systems," *Proc. ICASSP*, pp.I-189–192 (1992).

[86] Stephanie Seneff, Helen Meng and Victor Zue: "Language modelling for recognition and understanding using layered bigrams," *Proc. ICSLP 92*, pp.317–320 (1992).

[87] R. Pieraccini, E. Tzoukermann, Z. Gorelov, J.-L. Gauvain, E. Levin, C.H. Lee, and J.G. Wilpon: "A speech understanding system based on statistical representation of semantics," *Proc. ICASSP*, pp.I-193–196 (1992).

[88] 坪井宏之, 橋本秀樹, 竹林洋一：キーワードスポッティングに基づく連続音声理解, 信学技報, SP91-95 (1991).

[89] Y. Yamashita, H. Yoshida, T. Hiramatsu, Y. Nomura and R. Mizoguchi, "MASCOT II, A dialogue manager in general interface for speech input and output," *IEICE Trans. Inf. Syst.*, Vol.E76-D, No.1, pp.74–83 (1993).

[90] S. R. Young, A. G. Hauptmann, W. H. Ward, E. T. Smith and P. Werner, "High level knowledge sources in usable speech recognition system," *Commun. ACM*, Vol.32, No.2, pp.183–194 (1989).

[91] Douglas O'Shaughnessy: "Analysis and automatic recognition of false starts in spontaneous speech," *Proc. ICASSP*, pp.II-724–727 (1993).

[92] 中川聖一, 大黒慶久：連続音声認識・理解システムのための構文解析法の比較・検討, 情報処理学会論文誌, Vol.30, No.8, pp.932–943 (1989).

[93] A.V. Oppenheim, and D.H. Johnson: "Discrete representation of signals," Proc. IEEE, Vol.60, pp.681–691 (1972).

[94] 徳田, 小林, 今井: メル一般化ケプストラムの再帰的計算法, 信学論, Vol.71-A, No.1, pp.128–131 (1988).

信学会：電子情報通信学会

信学論：電子情報通信学会論文誌

信学技報：電子情報通信学会技術報告

ICASSP：International Conference on Acoustics, Speech, and Signal Processing

ICSLP：International Conference on Spoken Language Processing

IEEE：The Institute of Electrical and Electronics Engineers, Inc.

IEICE：The Institute of Electronics, Information and Communication Engineers（電子情報通信学会）

付録 A.

「UNIX-QA」タスクによる評価実験

3章の文音声認識実験での評価用音声データの発話文のリストと、3.8.9節の文法の記述単位に関する評価での補足的な実験結果を示す。

A.1 評価用 50 文リスト（電子メールに関する文）

（メール送信）

1. 現在作成したメールを田中さんへ送信してほしい
2. 鈴木氏にメールを送りたい
3. メールを暗号化して渡せるか?
4. ファイルの内容を送信したい

（メール受信）

5. メールの到着を知りたい
6. メールが来ていることを知るにはどうすればよいか?
7. メールを見たい
8. 次のメールを見るにはどのようにすればいいですか?
9. メールを消すにはどうすればいいのか?
10. 受信したメールをすべて表示するには?
11. 消去指示のない全部のメッセージを、メールファイルに保持したまま終われますか?
12. 指定したファイルにメッセージを保存したい
13. どうすれば表示中のメッセージを消してから、次のメッセージを表示できますか?
14. 表示しているメッセージをどうやれば再表示できるか?
15. 一つ前のメッセージをもう一度見たい
16. 指定したファイルにヘッダをつけずにメッセージを残すには?
17. 受信した全てのメッセージを変更せずに、メールファイルに保持したまま終了したい
18. 指示コマンド一覧をお願いします
19. 指示コマンドの全てを示して下さい
20. 受信したメールを到着順に処理していくには?
21. 次のメッセージを見る方法を示せ
22. メールを終わりたい

23. 最後からメッセージを見たい
24. メッセージを印刷するには?
25. 佐藤からのメールはありますか?
26. 暗号解読用キーを登録したい
27. 暗号化メールを見たい
28. どうやってメールが他人に読まれるのを防ぐのか?
29. 誰からのメールが来ているか?
30. メールコマンドのオプションを知りたい
31. メールコマンドから出てシェルに戻りたい
32. ファイルにメッセージは保存できますか?
33. メール変更処理を無効にしたい
34. メールが到着次第メールを表示したい

(メール転送)
35. メールを転送したい
36. 山本君からのメールを田中さんへ送れますか?

(メッセージ送信)
37. ログイン名近藤にメッセージを送る方法を教えてくれ
38. 特定の端末にメッセージを送りたい
39. ログインしている全てのユーザーへメッセージを送信したい
40. どうすれば指定した日付にメッセージを送ることができますか

(メッセージ受信禁止)
41. メッセージ受信を禁止してほしい
42. メッセージを受けたいのです
43. メッセージ受信の禁止を解除したいが
44. メッセージ受信の状態を知るには?
45. どの様にメッセージ受信を許可するのですか?

(その他)
46. 今日の日付を教えて下さい
47. 今の時刻を知りたい
48. 誰がログインしているか?
49. 端末番号を知りたい
50. 私は誰か?

A.2 文法の記述単位に関する比較・評価実験（補足）

ここでは、3.8.9節の実験に加えて、タスクの語彙数の影響を調べるために3.8.9節で使用した単語単位及び音節単位の文法の語彙数（単語数）を作為的に減らして実験を行なった結果を示す。まず、単語単位の文法でテストデータで使われていない単語の中から任意に選んだ単語を辞書から削除して新たな文法を作り、その文法を変換して

音節単位の文法も作成した。認識結果（表 A.2）を見ると、当然ながら認識時間は一様に短くなるものの、文法の記述単位の違いによる認識効率への影響については、3.8.9節での結果と同様にやや音節単位の文法の方法が効率が良い傾向が見られる。

表 A.1: 文法の規則数等の比較

文法の記述単位		単語単位	音節単位
終端記号数		313	83
非終端記号数		259	563
ワードクラス数		268	0
書換え規則数	主要部	534	927
	ワードクラス	349	0
テストセット perplexity（単語／音節）		7.5	2.3

表 A.2: 文認識率（%）

評価タスク：電子メール関係（語彙数を減らした文法 (表 A.1)）
評価データ： 全 281 文（6 名分）

条件： HMM 継続時間制御あり（3 乗して重み付け）
しきい値 $\lambda = -250$

(a) 単語単位で記述された文法 (単語 perplexity=7.5) の場合

ビーム幅	SN	TI	HU	KO	MA	SE	ALL	TIME[s]
3	93.8(1)	93.6	89.1	91.3(1)	91.8	95.6	92.5	84
5	93.8	93.6	91.3	91.3	93.9	95.6	93.2	95
10	97.9	93.6	91.3	91.3	93.9	95.6	94.0	98
20	97.9	93.6	91.3	91.3	93.9	95.6	94.0	102

(b) 音節単位で記述された文法 (音節 perplexity=2.3) の場合

ビーム幅	SN	TI	HU	KO	MA	SE	ALL	TIME[s]
10	83.3(4)	89.4	89.1	89.1(1)	91.8	93.3	89.3	77
20	93.8	91.5	91.3	91.3	93.9	95.6	92.9	83
40	93.8	93.6	91.3	91.3	93.9	95.6	93.2	85
80	95.8	93.6	91.3	91.3	93.9	95.6	93.6	94

（括弧内の数字は認識結果が出力されなかった文数）

付録 B.

「富士山観光案内」タスクによる評価実験

4章の未知語・冗長語の処理の評価実験でのシステムの文法と、評価用音声データの発話文リスト、間投詞リストなどを示す。付録 Bの最後には、本タスクで未知語・冗長語がないと仮定した条件でのシステム SPOJUS-SYNO-X を用いた認識実験結果を示す。

B.1 システムの文法

音声認識システムの言語制約として使用した文脈自由文法の書き換え規則を以下に示す。本文法は、冗長語に関する規則は含まれておらず、4章の評価実験で実際に用いる文法のベースとなるものである。以下の記述では、"A → B C" の書き換え規則を、"A B C" と記す。また、非終端記号を大文字の単語で、ワードクラス（前終端記号）を小文字の単語でそれぞれ表す。"/" の右側の単語は、3.3.1節で述べたユーザの次発話予測において用いる構文カテゴリ名を示している。

```
1: SSSS    SSSP

2: SSSP    PR01  PR02
3: PR01    PREA  PREB
4: PR02    PREC  SNT0
5: PR02
6: SNT0    SNTA
7: SNT0    SNTX  /  S_OREI

8: PREA    henji_yesno  /  S_YES_NO
9: PREA    henji_aisatu  /  S_AISATU
10: PREA
11: PREB   henji2
12: PREB
13: PREC   conj
14: PREC
15: SNTX   yorosiku
16: SNTX   please
17: SNTX   yorosiku  please
18: SNTX   doumo
19: SNTX   SANX
20: SNTX   doumo  SANX
21: SNTX   wakarimasita
22: SANX   thank  GOZA
23: GOZA   gozai
24: GOZA

25: SNTA   S000  SYUJOSI
26: SYUJOSI   syujosi
27: SYUJOSI

28: S000   S_IKU  /  S_IKU
29: S000   S_ARU  /  S_ARU
30: S000   S_KAKARU  /  S_KAKARU
31: S000   S_DESU  /  S_DESU
```

```
32: S000 S_NOBORU / S_NOBORU
33: S000 S_NORU / S_NORU
34: S000 S_HAIRU / S_HAIRU
35: S000 S_OSIETE / S_OSIETE
36: S000 S_RENRAKU / S_RENRAKU
37: S000 S_YORU / S_YORU
38: S000 S_SYUKUHAKU / S_SYUKUHAKU
39: S000 S_SITAI / S_SITAI
40: S000 S_ONEGAI / S_ONEGAI
41: S000 S_ADJ_OOKII / S_ADJ_OOKII
42: S000 S_ADJ_YUUMEI / S_ADJ_YUUMEI
43: S000 S_HITUYOO / S_HITUYOO
44: S000 S_ADJ_YOI / S_ADJ_YOI
45: S000 S_ADJ_OOI / S_ADJ_OOI
46: S000 S_KANGAERU / S_KANGAERU
47: S000 S_NARU / S_NARU
48: S000 S_MAWARU / S_MAWARU
49: S000 S_OBJ-TUKU / S_OBJ-TUKU
50: S000 S_TUKERU / S_TUKERU
51: S000 S_YOTEI / S_YOTEI
52: S000 S_MANIAU / S_MANIAU
53: S000 S_MIERU / S_MIERU
54: S000 S_KOOKAI / S_KOOKAI
55: S000 S_HARAU / S_HARAU
56: S000 S_RYOKOO / S_RYOKOO
57: S000 S_TOIAWASE / S_TOIAWASE
58: S000 S_KYOOMIARU / S_KYOOMIARU
59: S000 S_KANKOU / S_KANKOU

60: S000 S_PLACE_DESU / S_PLACE_DESU
61: S000 S_VEHICLE_DESU / S_VEHICLE_DESU
62: S000 S_BANGOU_DESU / S_BANGOU_DESU

63: S_KANKOU PLACE_IKU JOSI_WO kankou HUM_SURU

64: S_BANGOU_DESU BANGOU DESU_E
65: BANGOU num ban

66: S_PLACE_DESU PLACE_DESU JOSI_LOC_E DESU_E
67: PLACE_DESU PLACE_IKU
68: PLACE_DESU city_from ATARI_MAWARI_E
69: PLACE_DESU city ATARI_MAWARI_E
70: JOSI_LOC_E JOSI_TO_IKU
71: JOSI_LOC_E JOSI_FROM_IKU
72: JOSI_LOC_E
73: DESU_E desu
74: DESU_E

75: S_VEHICLE_DESU vehicle JOSI_BY_E DESU_E
76: JOSI_BY_E JOSI_DE
77: JOSI_BY_E

78: HUM_SURU hum_suru_no TO_OMOU_E
79: HUM_SURU hum_suru_no ndesu
80: HUM_SURU hum_suru_to TO_OMOU_E
81: HUM_SURU hum_suru_sy

82: S_KYOOMIARU koyu_spot fjosi_toiu tokoro JOSI_HA KYOOMIARU
83: KYOOMIARU kyoomi JOSI_HA ARU
84: JOSI_HA_E JOSI_HA
85: JOSI_HA_E

86: S_TOIAWASE TYOTTO_E RYOKOO_CONJ TOIAWASE_SURU
87: TYOTTO_E tyotto
88: TYOTTO_E
89: RYOKOO_CONJ ryokoo JOSI_WO_E hum_suru_no JOSI_NODE
90: RYOKOO_CONJ ryokoo JOSI_WO_E hum_suru_sy JOSI_NODE
91: TOIAWASE_SURU toiawase JOSI_WO_E HUM_SURU
92: JOSI_WO_E JOSI_WO
93: JOSI_WO_E

94: S_RYOKOO TOKI_RYOKOO_E PLACE_IKU JOSI_HE RYOKOO_SURU
95: TOKI_RYOKOO_E TUKI NO_SUE_E KURAI_E JOSI_NI
96: TOKI_RYOKOO_E
97: RYOKOO_SURU ryokoo HUM_SURU
98: NO_SUE_E JOSI_NO sue
99: NO_SUE_E

100: S_HARAU sonna tokoro JOSI_HA HARAU_OBJ JOSI_WO HARAU
101: HARAU_OBJ ryookin1 TOKA_E
102: HARAU_OBJ ryookin2 TOKA_E
103: HARAU harau_no TO_OMOU_E
104: HARAU harau_no ndesu
105: HARAU harau_to TO_OMOU_E
106: HARAU harau_sy

107: S_KOOKAI norm_place JOSI_HA KOOKAI_SARERU
108: KOOKAI_SARERU kookai OBJ_SARERU
109: KOOKAI_SARERU kookai OBJ_SURU
110: OBJ_SARERU obj_sareru_no TO_OMOU_E
111: OBJ_SARERU obj_sareru_no ndesu
112: OBJ_SARERU obj_sareru_sy
113: OBJ_SURU obj_suru_no TO_OMOU_E
114: OBJ_SURU obj_suru_no ndesu
115: OBJ_SURU obj_suru_sy

116: S_MIERU KEKKO_E TOOKU_E JOSI_MADE MIERU
117: TOOKU_E tooku
118: TOOKU_E
119: MIERU mieru_no TO_OMOU_E
120: MIERU mieru_no ndesu
121: MIERU mieru_sy

122: S_MANIAU ima JOSI_KARA yoyaku hum_suru_conj MANIAU
123: MANIAU maniau_no TO_OMOU_E
124: MANIAU maniau_no ndesu
125: MANIAU maniau_sy

126: S_YOTEI nissu KURAI_E YOTEI_SURU
127: YOTEI_SURU yotei HUM_SURU

128: S_OBJ-TUKU syokuzi TOKA_E JOSI_HA OBJ-TUKU
129: OBJ-TUKU obj-tuku_no TO_OMOU_E
130: OBJ-TUKU obj-tuku_no ndesu
131: OBJ-TUKU obj-tuku_sy

132: S_TUKERU syokuzi TOKA_E JOSI_HA TUKERU
133: TUKERU tukeru_no TO_OMOU_E
134: TUKERU tukeru_no ndesu
135: TUKERU tukeru_to TO_OMOU_E
136: TUKERU tukeru_sy

137: S_MAWARU MAWARU_PLACES issyoni mawaru_np JOSI_NO JOSI_HA YOIDESU
138: S_MAWARU sore JOSI_HA ZYUUBUN_E MITEMAWARU
139: S_MAWARU norm_build JOSI_HA JIKAN_DE MITEMAWARU
140: S_MAWARU ATO_E KESIKI JOSI_WO IKUTUKA_E MITEMAWARU
141: MAWARU_PLACES KOYU_PLACE JOSI_AND KOYU_PLACE JOSI_WO
142: ATO_E ato
143: ATO_E
144: KESIKI kesiki JOSI_NO yoi TOKORO
145: KESIKI kesiki JOSI_HA kireina TOKORO
146: IKUTUKA_E ikutu JOSI_KA
147: IKUTUKA_E
148: JIKAN_DE DAITAI_E HOURS KURAI_E JOSI_DE
149: ZYUUBUN_E zyuubun
150: ZYUUBUN_E zyuubun
151: MITEMAWARU MIRU_CONJ_E mawaru_no TO_OMOU_E
152: MITEMAWARU MIRU_CONJ_E mawaru_no ndesu
```



```
153: MITEMAWARU MIRU_CONJ_E mawaru_to TO_OMOU_E
154: MITEMAWARU MIRU_CONJ_E mawaru_sy

155: MIRU_CONJ_E miru_conj
156: MIRU_CONJ_E

157: S_NARU PLACE_NARU1 soko JOSI_NI NARU
158: S_NARU PLACE_NARU2 PLACE_NARU3 mukatte NANIGAWA JOSI_NI NARU
159: PLACE_NARU1 PLACE_NARUS JOSI_NI tikai norm_place JOSI_HA
160: PLACE_NARU2 PLACE_NARUS fjosi_toiu JOSI_NO JOSI_HA
161: PLACE_NARU3 koyu_mountain JOSI_NO norm_place JOSI_KARA
162: PLACE_NARUS KOYU_PLACE
163: NARU naru_no TO_OMOU_E
164: NARU naru_no ndesu
165: NARU naru_sy
166: NANIGAWA nani gawa
167: NANIGAWA hookoo gawa

168: S_KANGAERU ippaku kurai JOSI_DE KANGAERU
169: S_KANGAERU holiday JOSI_NO koro JOSI_WO KANGAERU
170: S_KANGAERU NINZUU kurai JOSI_WO KANGAERU
171: KANGAERU kangaeru_no TO_OMOU_E
172: KANGAERU kangaeru_no ndesu
173: KANGAERU kangaeru_to TO_OMOU_E
174: KANGAERU kangaeru_sy

175: S_ADJ_OOI OOI_PLACE1 OOI_PLACE2 OOIDESU
176: OOI_PLACE1 TOMARU tokoro JOSI_HA
177: OOI_PLACE2 dono norm_place ATARI_MAWARI_E JOSI_HA
178: OOIDESU ooi adj_desu

179: S_ADJ_YOI dono atari JOSI_HA YOIDESU
180: S_ADJ_YOI sotira JOSI_NI dooyatte iku_conj YOIDESU
181: S_ADJ_YOI YOYAKUJI doositara YOIDESU
182: YOYAKUJI yoyaku JOSI_NO toki JOSI_NI
183: YOIDESU yoi adj_desu

184: S_HITUYOO TOKUNI_E yoyaku TOKA_E JOSI_HA_E HITUYOO_DESU
185: HITUYOO_DESU hituyoo desu
186: HITUYOO_DESU hituyoo ARU
187: TOKUNI_E tokuni
188: TOKUNI_E

189: S_ADJ_YUUMEI DESU_SUBJ norm_place JOSI_HA yuumei desu

190: S_ADJ_OOKII PLACE_OOKII KEKKO_E OOKIIDESU
191: S_ADJ_OOKII ATO_E PLACE_OOKII dore JOSI_HA itiban OOKIIDESU
192: S_ADJ_OOKII ATO_E dore JOSI_HA itiban NIGIYAKADESU
193: KEKKO_E dono kurai
194: KEKKO_E dore kurai
195: KEKKO_E kekko
196: KEKKO_E
197: PLACE_OOKII koyu_spot JOSI_HA
198: PLACE_OOKII norm_place JOSI_HA
199: OOKIIDESU ookii adj_desu
200: NIGIYAKADESU nigiyaka desu

201: S_ONEGAI yoyaku JOSI_HA sotira JOSI_DE ONEGAI
202: S_ONEGAI SYUKUHAKU_PLACE tekitouna tokoro JOSI_WO ONEGAI
203: S_ONEGAI soko JOSI_NI NINZUU JOSI_DE ONEGAI
204: S_SITAI syukuhaku JOSI_HA norm_build JOSI_NI HUM_SURU
205: S_SITAI soko JOSI_WO ONEGAI

206: ONEGAI onegai HUM_SURU

207: S_SYUKUHAKU PLACE_IKU JOSI_DE TOMARU
208: S_SYUKUHAKU SYUKUHAKU_PLACE TOMARU
209: S_SYUKUHAKU MONTH_DAY TOMARU
210: SYUKUHAKU_PLACE koyu_place SYUKUHAKU_PLACE2_E ATARI_MAWARI_E JOSI_DE
211: SYUKUHAKU_PLACE2_E JOSI_AND koyu_place
212: SYUKUHAKU_PLACE2_E
213: MONTH_DAY TUKI JOSI_NO hiniti JOSI_NI

214: S_YORU KESIKI JOSI_NI YORU
215: YORU yoru_no TO_OMOU_E
216: YORU yoru_no ndesu
217: YORU yoru_to TO_OMOU_E
218: YORU yoru_sy

219: S_RENRAKU GUTAITEKINI_E kimattara MATA_E sotira JOSI_NI RENRAKU
220: MATA_E mata
221: MATA_E
222: RENRAKU renraku HUM_SURU

223: S_OSIETE GUTAITEKINI_E MIDOKORO_GA aru_no JOSI_ARUKA OSIETE
224: S_OSIETE ATO_E KESIKI JOSI_WO IKUTUKA_E OSIETE
225: GUTAITEKINI_E gutaitekini
226: GUTAITEKINI_E
227: MIDOKORO_GA DONNA midokoro JOSI_HA
228: JOSI_ARUKA JOSI_KA
229: JOSI_ARUKA JOSI_NOKA
230: OSIETE osiete_no TO_OMOU_E
231: OSIETE osiete_no ndesu
232: OSIETE osiete_to TO_OMOU_E
233: OSIETE osiete_sy

234: S_HAIRU OSORAKU_E PLACE_HAIRU VEHICLE_DE HAIRU
235: S_HAIRU PLACE_HAIRU DONNA VEHICLE_DE HAIRU
236: OSORAKU_E osoraku
237: OSORAKU_E
238: VEHICLE_DE vehicle JOSI_DE
239: PLACE_HAIRU soko JOSI_NI
240: PLACE_HAIRU koyu_spot fjosi_toiu JOSI_NO JOSI_HA
241: HAIRU hairu_no TO_OMOU_E
242: HAIRU hairu_no ndesu
243: HAIRU hairu_to TO_OMOU_E
244: HAIRU hairu_sy

245: S_NORU vehicle JOSI_NI NORU
246: NORU noru_no TO_OMOU_E
247: NORU noru_no ndesu
248: NORU noru_to TO_OMOU_E
249: NORU noru_sy

250: S_NOBORU ASAHAYAKU_E koyu_mountain JOSI_NI VEHICLE_DE_E NOBORU
251: S_NOBORU dono atari JOSI_MADE VEHICLE_DE_E NOBORU TO_OMOU_E
252: NOBORU tozan HUM_SURU
253: NOBORU noboru_no TO_OMOU_E
254: NOBORU noboru_no ndesu
255: NOBORU noboru_to TO_OMOU_E
256: NOBORU noboru_sy
257: VEHICLE_DE_E VEHICLE_DE
258: VEHICLE_DE_E
259: ASAHAYAKU_E asahayaku
260: ASAHAYAKU_E
```

```
261: S_DESU sakihodo JOSI_NO DEGREE kurai JOSI_DE KEKKODESU
262: S_DESU DONNA tokoro JOSI_HA midokoro desu
263: S_DESU ATO_E itiban ookii DESU_SUBJ sore desu
264: S_DESU sono vehicle JOSI_HA vehicle desu
265: S_DESU DESU_SUBJ nan fjosi_toiu tokoro desu
266: S_DESU DESU_SUBJ vehicle JOSI_DE hairu_no tokoro desu
267: S_DESU TORIAEZU_E sore KURAI_E JOSI_DE KEKKODESU
268: KEKKODESU kekko desu
269: TORIAEZU_E toriaezu
270: TORIAEZU_E
271: S_DESU KOYU_PLACE JOSI_NO hoo JOSI_HA doo desu
272: S_DESU DESU_SUBJ DESU_PRED desu
273: DESU_SUBJ PLACE_DESU fjosi_toiu JOSI_TO_IF
274: DESU_SUBJ PLACE_DESU fjosi_toiu JOSI_NO JOSI_HA
275: PLACE_DESU KOYU_PLACE
276: PLACE_DESU norm_spot
277: DESU_PRED itutu JOSI_NO norm_place
278: DESU_PRED norm_place JOSI_HA itutu aru_no fjosi_toiu koto

279: KOYU_PLACE koyu_place
280: KOYU_PLACE koyu_mountain
281: KOYU_PLACE koyu_spot

282: S_ARU IN_LOC_E ARU_OBJ_GA ARU
283: S_ARU norm_build dehanakute ARU_OBJ_GA ARU
284: S_ARU ARU_OBJ_GA IN_LOC ARU
285: S_ARU ARU_LOC_DE_E NANI JOSI_KA ARU
286: NANI nan
287: NANI nani
288: ARU aru_no TO_OMOU_E
289: ARU aru_no ndesu
290: ARU aru_sy

291: IN_LOC_E IN_LOC
292: IN_LOC_E

293: IN_LOC PLACE_IKU JOSI_NI
294: IN_LOC PLACE_IKU JOSI_NI
295: IN_LOC DAITAI_E sotira gawa JOSI_NI
296: IN_LOC syukuhaku JOSI_NI
297: IN_LOC ippaku JOSI_NI

298: ARU_OBJ_GA DONNA_E ARU_OBJ TOKA_E AND_ARU_OBJ_E JOSI_HA
299: ARU_OBJ vehicle
300: ARU_OBJ norm_build
301: ARU_OBJ KOYU_PLACE
302: ARU_OBJ norm_spot
303: ARU_OBJ norm_place
304: ARU_OBJ SYOKUZISURU_E TOKORO
305: ARU_OBJ nani
306: AND_ARU_OBJ_E JOSI_AND ARU_OBJ TOKA_E
307: AND_ARU_OBJ_E

308: SYOKUZISURU_E syokuzi obj_suru_no
309: SYOKUZISURU_E
310: TOKORO tokoro
311: TOKORO mono

312: DONNA_E DONNA
313: DONNA_E

314: ARU_LOC_DE_E PLACE_IKU JOSI_DE TOKUNI_E kankou fjosi_toiu JOSI_TO_IF

315: S_KAKARU KAKARU_OPTION_E RYOOKIN_HA_E DAITAI_E DONO_KURAI KAKARU

316: RYOOKIN_HA_E DEGREE TOKA_E JOSI_HA
317: RYOOKIN_HA_E KAKARU_OBJ JOSI_NO DEGREE TOKA_E JOSI_HA
318: RYOOKIN_HA_E KAKARU_OBJ JOSI_HA
319: RYOOKIN_HA_E
320: KAKARU_OBJ vehicle
321: KAKARU_OBJ norm_build
322: KAKARU_OBJ koyu_spot
323: KAKARU_OBJ norm_spot
324: KAKARU_OBJ norm_place
325: KAKARU_OBJ syukuhaku
326: KAKARU_OBJ ippaku
327: KAKARU_OBJ action
328: DEGREE ryookin1
329: DEGREE ryookin2
330: DEGREE time
331: DAITAI_E daitai
332: DAITAI_E
333: DONO_KURAI dono kurai
334: DONO_KURAI dore kurai
335: DONO_KURAI ikura KURAI_E
336: DONO_KURAI HOURS KURAI_E
337: DONO_KURAI kekko
338: KURAI_E kurai
339: KURAI_E
340: HOURS nan hour
341: HOURS nan minute
342: TOKA_E toka
343: TOKA_E JOSI_NO hoo
344: TOKA_E fjosi_toiu JOSI_NO
345: TOKA_E

346: KAKARU kakaru_no TO_OMOU_E
347: KAKARU kakaru_no ndesu
348: KAKARU kakaru_sy
349: KAKARU JOSI_NI_NARU NARU
350: KAKARU hituyoo desu
351: KAKARU desu

352: KAKARU_OPTION_E FROM_PLACE DATO_E
353: KAKARU_OPTION_E TO_PLACE DATO_E
354: KAKARU_OPTION_E soko JOSI_HA issyu obj_suru_no josi_noni
355: KAKARU_OPTION_E sore JOSI_HA
356: KAKARU_OPTION_E norm_build JOSI_NI TOMARU JOSI_TO_IF
357: KAKARU_OPTION_E sotira TOKA_E JOSI_HA
358: KAKARU_OPTION_E

359: DATO_E JOSI_DATO
360: DATO_E JOSI_HA
361: DATO_E

362: TOMARU tomaru
363: TOMARU syukuhaku HUM_SURU
364: TOMARU ippaku HUM_SURU

365: S_IKU ADVS_IKU_E SUBJ_IKU IKU
366: S_IKU JUNBIS IKU

367: ADVS_IKU_E ADVS_IKU
368: ADVS_IKU_E
369: ADVS_IKU ima
370: ADVS_IKU DAYS

371: JUNBIS KOTIRADE_E yoyaku JOSI_WO totte JOSI_KARA
372: KOTIRADE_E kotira JOSI_DE
373: KOTIRADE_E

374: JUNBIS ADVS_IKU_E WITH_IKU
375: WITH_IKU tomodati JOSI_WITH NINZUU JOSI_WITH_DE

376: IKU IKU_SY
377: IKU IKU_NO TO_OMOU_E NODE_E
378: IKU IKU_TO JOSI_TO OMOU NODE_E
379: IKU_NO RYOKOO_NI_E iku_no
380: IKU_TO RYOKOO_NI_E iku_to
381: IKU_SY RYOKOO_NI_E IKU_SY2
```



```
382: IKU_SY2 iku_sy
383: IKU_SY2 iku_no ndesu
384: IKU_SY2 iku_no yotei desu

385: NODE_E JOSI_NODE totyuu JOSI_MADE IKU_SY
386: NODE_E

387: RYOKOO_NI_E ryokoo JOSI_RYOKOO_NI
388: RYOKOO_NI_E

389: SUBJ_IKU FROM_PLACE TO_PLACE_E BY_VEHICLE_E
390: SUBJ_IKU TO_PLACE BY_VEHICLE_E
391: SUBJ_IKU BY_VEHICLE_E FROM_PLACE_E TO_PLACE_E

392: FROM_PLACE_E FROM_PLACE
393: FROM_PLACE_E

394: TO_PLACE_E TO_PLACE
395: TO_PLACE_E

396: BY_VEHICLE_E BY_VEHICLE
397: BY_VEHICLE_E

398: FROM_PLACE PLACE_IKU JOSI_FROM_IKU
399: FROM_PLACE city_from ATARI_MAWARI_E JOSI_FROM_IKU
400: TO_PLACE PLACE_IKU JOSI_TO_IKU
401: TO_PLACE city ATARI_MAWARI_E JOSI_TO_IKU
402: BY_VEHICLE VEHICLE_IKU JOSI_BY_IKU

403: PLACE_IKU KOYU_PLACE ATARI_MAWARI_E
404: PLACE_IKU norm_build ATARI_MAWARI_E
405: PLACE_IKU norm_spot ATARI_MAWARI_E
406: PLACE_IKU norm_place ATARI_MAWARI_E
407: PLACE_IKU soko ATARI_MAWARI_E
408: PLACE_IKU sono ATARI_MAWARI
409: PLACE_IKU sotira

410: ATARI_MAWARI JOSI_NO_E atari
411: ATARI_MAWARI JOSI_NO mawari

412: ATARI_MAWARI_E ATARI_MAWARI
413: ATARI_MAWARI_E

414: VEHICLE_IKU vehicle
415: VEHICLE_IKU NANI

416: JOSI_NO_E JOSI_NO
417: JOSI_NO_E

418: JOSI_FROM_IKU JOSI_KARA
419: JOSI_TO_IKU JOSI_HE
420: JOSI_TO_IKU JOSI_NI
421: JOSI_TO_IKU JOSI_MADE
422: JOSI_BY_IKU JOSI_DE

423: TO_OMOU_E JOSI_TO omou_no NDESU_E
424: TO_OMOU_E JOSI_TO omou_sy
425: TO_OMOU_E JOSI_TO kangaeru_no NDESU_E
426: TO_OMOU_E JOSI_TO kangaeru_to
427: TO_OMOU_E JOSI_TO kangaeru_sy
428: TO_OMOU_E

429: NDESU_E ndesu
430: NDESU_E

431: OMOU omou_no
432: OMOU omou_sy

433: DAYS TUGINO_E HOLIDAY JOSI_NI

434: TUGINO_E tugi JOSI_NO
435: TUGINO_E

436: HOLIDAY holiday ATARI_DAY_E
437: ATARI_DAY_E atari_day
438: ATARI_DAY_E

439: NINZUU ninzuu

440: TUKI tuki_num gatu

441: JOSI_WO josi_wo
442: JOSI_DE josi_de
443: JOSI_HA josi_ha
444: JOSI_NODE josi_node
445: JOSI_HE josi_he
446: JOSI_NI josi_ni
447: JOSI_NO josi_no
448: JOSI_KARA josi_kara
449: JOSI_MADE josi_made
450: JOSI_KA josi_ka
451: JOSI_AND josi_and
452: JOSI_TO josi_to
453: JOSI_WITH josi_with
454: JOSI_WITH_DE josi_with_de
455: JOSI_NI_NARU josi_ni_naru
456: JOSI_TO_IF josi_to_if
457: JOSI_RYOKOO_NI josi_ryokoo_ni
458: JOSI_DATO josi_dato
459: JOSI_NOKA josi_noka

460: DONNA donna
```

B.2 冗長語を許した文法

前節の文法では間投詞のワードクラスが使用されておらず、間投詞が含まれる文を受理できない。4.4.4節の間投詞を含む発話の認識実験では、冗長語が文頭や接続詞の後ろ、助詞の後ろに出現可能とするために、前節の書き換え規則の一部を変更して使用している。ここでは冗長語の挿入を許すために変更及び追加された書き換え規則のみを示す。

```
  2: SSSP    KANT_E  PR01  PR02
  8: PREA    henji_yesno  KANT_E  /  S_YES_NO
  9: PREA    henji_aisatu  KANT_E  /  S_AISATU
 11: PREB    henji2  KANT_E
 13: PREC    conj  KANT_E
441: JOSI_WO    josi_wo  KANT_E
442: JOSI_DE    josi_de  KANT_E
443: JOSI_HA    josi_ha  KANT_E
444: JOSI_NODE    josi_node  KANT_E
445: JOSI_HE    josi_he  KANT_E
446: JOSI_NI    josi_ni  KANT_E
447: JOSI_NO    josi_no  KANT_E
448: JOSI_KARA    josi_kara  KANT_E
449: JOSI_MADE    josi_made  KANT_E
450: JOSI_KA    josi_ka  KANT_E
451: JOSI_AND    josi_and  KANT_E
452: JOSI_TO    josi_to  KANT_E
453: JOSI_WITH    josi_with  KANT_E
454: JOSI_WITH_DE    josi_with_de  KANT_E
455: JOSI_NI_NARU    josi_ni_naru  KANT_E
456: JOSI_TO_IF    josi_to_if  KANT_E
457: JOSI_RYOKOO_NI    josi_ryokoo_ni  KANT_E
458: JOSI_DATO    josi_dato  KANT_E
459: JOSI_NOKA    josi_noka  KANT_E
460: DONNA    donna  KANT_E
461: KANT_E    kant
462: KANT_E
```

最後の 2 行は新たに追加された書き換え規則で、“kant” は冗長語（間投詞）を代表するワードクラスである。4.4.4節の実験では、1) 付録 B.4の間投詞リストの一部の単語をこのワードクラスに登録する、2) このワードクラスの単語を完全に未知語として処理する、3) 間投詞の単語登録と未知語処理を併用する、という方法で評価している。

B.3 評価用 104 文リスト

001:富士山周辺で特に観光というと、何(naN)か有りますかね。
002:富士五湖というのはだいたいどちら側にあるんですか。
003:富士五湖っていうと五つの湖ですよね。
004:富士五湖にはどんな観光地があるんですか。
005:そうですか。山中湖の方はどうですかね。
006:遊覧船はだいたい何分ぐらいでまわれるんですかね。
007:そうですか。それで、富士五湖の周辺に富士急ハイランドとかサファリパークとかあるわけですか。

008:富士急ハイランドは、結構大きいんですかね。
009:そちらの方は、入園料はどの程度になっているんですかね。
010:そうですか。そうなると、富士五湖と富士急ハイランドを一緒にまわるのがいいですかね。
011:そうですかね。それじゃ、どうもありがとうございました。
012:もしもし。
013:富士山に登山したいと思うんですけども。
014:車でも行けますよね。
015:どの辺まで車で登山できるんでしょうか。
016:そこには駐車場とかはありますか。
017:では、朝早く富士山に車で登りたいんですけども。
018:山中湖からだと、何時間ぐらいかかるんでしょうか。

019:そうですか。それで、五合目には食事するところとかあるんでしょうか。
020:わかりました。それでは、富士五湖の周辺に美術館とかはありますか。
021:そうですか。じゃあ、博物館とかはありますか。
022:そうですか。そこはどのようなものがあるんでしょうか。
023:そうですか。入場料とかはいくらくらいですか。
024:そうですか。湖には遊覧船とかはありますか。
025:遊覧船はいくらくらいですか。
026:はい、わかりました。ありがとうございました。
027:3月の末ぐらいに富士山の方へ旅行したいんですけど。
028:具体的にどういうみどころがあるのか教えてほしいんですけれども。
029:富士五湖というのは、湖が五つあるということですか。
030:そうですか。遊園地というのは、なんという場所ですか。
031:そうですか、サファリパークというと、車で入っていくところですね。
032:そうですか。それは入園料というのはどのくらい必要なんですか。
033:そうですか。おそらくそこにバスで入れると思うんですけれども。
034:そうすると、どのくらいかかるんでしょうか。
035:そうですか。特に予約とか必要ないですよね。
036:そうですか。あと、いちばん大きい遊園地というとどれですか。
037:そうですか。入園料はどのくらいかかりますか。
038:そうなんですか。で、そこには観覧車とかありますかね。
039:じゃあ、結構遠くまで見えるわけですね。
040:わかりました。どうもありがとうございました。
041:富士山というと風穴(fuuketu)が有名なんですけれども。
042:風穴は一般公開されているんですか。
043:そうですか。そういうところは、入園料とかを払うわけですよね。
044:そうですか。富士急ハイランドに近い風穴はどこになりますか。
045:そうですか。西湖というのは、富士山の頂上からむかって何側になりますか。
046:そうですか。じゃ、あと、湖はどれがいちばん大きいんですか。
047:そうですか。泊まるところは、どの湖のあたりが多いですかね。
048:そうですか。宿泊施設はどういうところがありますか。
049:そうですか、わかりました。
050:じゃ具体的に決まったら、またそちらに連絡したいと思います。
051:はい、じゃよろしくお願いします。
052:ちょっと旅行をしたいので、問い合わせをしたいんですが。
053:富士山の方へ行きたいと思っているんですけれども。
054:一泊ぐらいで考えています。
055:どういったところが見所でしょうか。
056:車で行こうと思いますので、途中まで行きます。
057:ですが、頂上までは行かない予定です。
058:はい、わかりました。景色のいいところを教えてもらおうと思います。
059:はい。でもサファリパークというところは興味がありますね。
060:サファリパークというのはどんな車でも入れるんですか。
061:そうですか。料金は、どれぐらいかかるんでしょうか。
062:そうですか。そこは、一周するのに時間は結構かかるんでしょうか。
063:それでは、あと景色のいいところをいくつかまわりたいと思います。

064:どのあたりがいいでしょうか。
065:河口湖か山中湖のあたりで宿泊しようと思っています。
066:宿泊はどんなところがあるでしょうか。
067:ペンションとかはないんでしょうか。
068:そうですか。宿泊はペンションにしたいと思います。
069:宿泊の料金は、だいたいいくらぐらいなんでしょうか。
070:そうですか。ゴールデンウィークのころを考えているんですけれども。
071:今から予約しても間に合うでしょうか。
072:３人ぐらいを考えています。
073:わかりました。では、予約の時にはどうしたらよいでしょうか。
074:先ほどの料金ぐらいでけっこうですけど。
075:そうですか。では、そこをお願いしたいと思います。
076:４月の３０日に宿泊しようと思います。
077:はい、そうです。
078:はい、よろしくおねがいします。
079:今度の連休に富士山方面に旅行に行きたいと思うんですけれど。
080:どういった見所があるか教えてください。
081:いえ、電車で行こうかと思っています。
082:そうですね。景色がきれいなところには、寄ってみたいと思います。
083:そうですか。で、二日ぐらい予定しているんですけども。
084:それらは十分見てまわれるんでしょうか。
085:そうですか。いま豊橋から電車で行こうと思っています。
086:そちらにはどうやって行けばいいでしょうか。
087:そうですか。新幹線に乗ろうと思っているんですけど。
088:新富士(siNfuji)まで値段はどれくらいかかりますか。
089:そうですか。時間の方はどれくらいかかりますか。
090:そうですか。そのバスはワンマンバスなんですか。
091:わかりました。とりあえずこれぐらいで結構です。
092:どうもありがとうございました。
093:富士五湖のあたりで一泊したいと考えてます。
094:どういった宿泊施設がありますか。
095:そうですね。ペンションではなくて温泉はないんですか。
096:じゃあ、ペンションなら、その近くにあるんですね。
097:ではペンションに泊まると、費用はどれくらいになりますか。
098:そうですか。食事とかはついているんでしょうか。
099:では、こっちで予約を取ってから行きたいと思います。
100:予約はそちらでお願いできるでしょうか。
101:では、再来週(saraisyu-)の週末あたりに友達と３人で行こうと思います。
102:そうですね。富士五湖のあたりで、適当なところをお願いしたいんですけども。
103:結構ですけど。
104:わかりました。じゃあ、そこに３人でお願いします。

B.4 間投詞リスト

音響学会連続音声データベースの音声対話書き起こしテキストの一部に現れた間投詞のリストを示す。調査対象の対話データは、対話数19、話者数23人の総文数1,769文である[69]。間投詞のラベルが付いている単語を抜きだし、連続して現れているもの(例えば、[えーとあのー])は適当な単位で分割した結果、総間投詞数は2,321個となった。観測された全間投詞(78種類)の頻度付きリストを以下に示す。

1	eeto	365	27	uuNto	4	54	sorede	1
2	ano	297	28	to	4	55	sooka	1
3	ee	293	29	iya	4	56	soo	1
4	a	236	30	N	4	57	soko	1
5	maa	178	31	uN	3	58	siee	1
6	anoo	163	32	soodesunee	3	59	ni	1
7	ma	158	33	sonoo	3	60	nani	1
8	e	142	34	eetoo	3	61	naaNka	1
9	aa	96	35	are	3	62	naNdesyoone	1
10	eto	79	36	site	2	63	naNdesukeredomo	1
11	eetodesune	48	37	naNte	2	64	naNdaroo	1
12	soodesune	38	38	naNdesuka	2	65	moo	1
13	sono	23	39	kono	2	66	koo	1
14	haa	21	40	hora	2	67	huuNto	1
15	NN	19	41	eee	2	68	huuN	1
16	etoo	17	42	aato	2	69	humu	1
17	kou	11	43	Nto	2	70	hee	1
18	zya	8	44	zyaa	1	71	hahaa	1
19	ato	8	45	yuuNdesuka	1	72	eetone	1
20	uuN	7	46	uuNtosa	1	73	eeeto	1
21	etodesune	6	47	uu	1	74	dee	1
22	eetodesunee	6	48	tyoto	1	75	atoo	1
23	naNka	5	49	tyootone	1	76	aree	1
24	naNda	5	50	tyau	1	77	aredesune	1
25	haaa	5	51	tuu	1	78	anosa	1
26	ha	5	52	too	1			
			53	sukosi	1			

B.5 間投詞入り評価用 50 文リスト

・含まれる間投詞 10 種

えと	/eto/
えーと	/eeto/
あの	/ano/
あのー	/anoo/
え	/e/
えー	/ee/
あ	/a/
あー	/aa/
ま	/ma/
まー	/maa/

001:[えーと] 富士山周辺で特に観光というと、何(naN)か有りますかね。
002:[えーと] 富士五湖というのはだいたいどちら側にあるんですか。
003:[えーと] 富士五湖っていうと五つの湖ですよね。
004:[えーと] 富士五湖にはどんな観光地があるんですか。
006:[えーと] 遊覧船はだいたい何分ぐらいでまわれるんですかね。
008:[あの] 富士急ハイランドは、結構大きいんですかね。
009:[あの] そちらの方は、入園料はどの程度になっているんですかね。
013:[あの] 富士山に登山したいと思うんですけども。
014:[あの] 車でも行けますよね。
015:[えー] どの辺まで車で登山できるんでしょうか。
016:[えー] そこには駐車場とかはありますか。
017:[えー] では、朝早く富士山に車で登りたいんですけども。
018:[えー] 山中湖からだと、何時間ぐらいかかるんでしょうか。
025:[あ] 遊覧船はいくらくらいですか。
027:[ま] 3月の末ぐらいに富士山の方へ旅行したいんですけど。
028:[あ] 具体的にどういうみどころがあるのか教えてほしいんですけれども。
029:[まー] 富士五湖というのは、湖が五つあるということですか。
039:[あ] じゃあ、結構遠くまで見えるわけですね。
041:[まー] 富士山というと風穴(fuuketu)が有名なんですけれども。
042:[あのー] 風穴は一般公開されているんですか。
050:[あ] じゃ具体的に決まったら、またそちらに連絡したいと思います。
051:[ま] はい、じゃよろしくお願いします。
052:[え] ちょっと旅行をしたいので、問い合わせをしたいんですが。
053:[あー] 富士山の方へ行きたいと思っているんですけれども。
054:[ま] 一泊ぐらいで考えています。

055:どういったところが[あの]見所でしょうか。
056:車で行こうと思いますので、[えーと]途中まで行きます。
057:ですが、[えーと]頂上までは行かない予定です。
060:サファリパークというのは[えーと]どんな車でも入れるんですか。
063:それでは、[えーと]あと景色のいいところをいくつかまわりたいと思います。

064:どのあたりが[えーと]いいでしょうか。
065:河口湖か[あの]山中湖のあたりで宿泊しようと思っています。
066:宿泊は[あの]どんなところがあるでしょうか。
067:ペンションとかは[あの]ないんでしょうか。
069:宿泊の料金は、[あの]だいたいいくらぐらいなんでしょうか。
071:今から[えー]予約しても間に合うでしょうか。
072:3人ぐらいを[えー]考えています。
074:先ほどの[えー]料金ぐらいでけっこうですけど。
076:4月の[えー]30日に宿泊しようと思います。
078:はい、[あ]よろしくおねがいします。
079:今度の連休に[あ]富士山方面に旅行に行きたいと思うんですけれど。
080:どういった[あ]見所があるか教えてください。
081:いえ、[あ]電車で行こうかと思っています。
084:それらは[まー]十分見てまわれるんでしょうか。
086:そちらには[まー]どうやって行けばいいでしょうか。
088:新富士(siNfuji)まで[あのー]値段はどれくらいかかりますか。
093:富士五湖のあたりで[あのー]一泊したいと考えてます。
094:どういった[ま]宿泊施設がありますか。
096:じゃあ、[ま]ペンションなら、その近くにあるんですね。
099:では、[あー]こっちで予約を取ってから行きたいと思います。

B.6 評価用 104 文の文認識実験結果

未知語や冗長語が含まれない場合についての SPOJUS-SYNO-X（未知語処理なし。3.5節参照）による認識実験結果を示す。システムの文法は、未知語処理の評価実験に用いたもののベースとなっている付録B.1の文法である。評価用データも、同じように未知語処理の評価実験に用いたもので、付録B.3に示した 104 文の 6 名の男性による発話である。単語単位のパープレキシティは 29.3 である。表B.1に、3.8.5節 (c) で述べた N-best 文認識法による第 5 位までの文認識率[1]を示す。第二位以下の候補まで見ると認識誤りの文数がかなり減少しているのが分かるが、細かい言い回しの違いなどの意味的に小さな間違いが多いことがおもに影響している。従って、このような N-Best 法を利用することの効果は、認識レベルで利用する言語知識の制約の強さに依存するものといえる。

表B.2は、文認識結果を単語単位の認識精度で評価した結果である。SEG.RATE はセグメンテーション率で、次式で計算される。

$$\text{SEG.RATE} = \frac{\text{入力単語数} - \text{脱落数} - \text{挿入数}}{\text{入力単語数}} \tag{B.1}$$

[1]文中の全単語が正しく認識されている文数の割合

表 B.1: N-Best 文認識の結果 (beam search 幅= 20)

順位	AK	MM	SA	TK	TS	YM	平均	誤認識文総数
1 位のみ	84.6	72.1	61.5	77.9	83.7	83.7	77.2	142
1〜2 位	93.3	88.5	73.1	86.5	90.4	92.3	87.3	79
1〜3 位	94.2	94.2	81.7	89.4	92.3	96.2	91.3	54
1〜4 位	94.2	96.2	88.5	90.4	94.2	96.2	93.3	42
1〜5 位	94.2	96.2	89.4	92.3	96.2	97.1	94.2	36

表 B.2: 文認識結果を単語単位で評価した結果

話者	入力単語数	正解 (%)	置換 (%)	挿入 (%)	脱落 (%)	SEG.RATE
AK	862	98.03	1.74	0.23	0.23	99.54
MM	863	96.64	3.01	0.70	0.35	98.96
SA	860	93.84	5.35	0.12	0.81	99.07
TK	862	96.64	3.13	0.46	0.23	99.30
TS	863	96.87	2.78	0.58	0.35	99.07
YM	862	98.03	1.74	0.12	0.23	99.65
平均	5172	96.67	2.96	0.37	0.37	99.27

付録 C.

リジェクションの評価実験

5章の孤立単語認識におけるリジェクションの評価に関連した補足的な情報と、文音声認識におけるリジェクション性能の評価実験に用いた評価用音声データの発話文リストを示す。

C.1 数値計算によるシミュレーション評価

5.2.2節のシミュレーションの仮定において、近似的に数値計算でリジェクション性能を求める方法を以下に示す。但し、ここで述べる計算法は、後に示す式で正解単語のスコア分布 $f(x)$ と未知語のスコア分布 $h(x)$ の相関を無視しているため（実際には、例えば図 5.1において、正解音節のスコア “232” が両者共通に用いられる必要がある）、登録単語の入力に対する単語認識率 Pc の計算式がシミュレーションの仮定と若干異なったものとなる。

初めに未知単語として認識される音節列のスコアの分布を考えると、各音節のスコア（確率変数を Y とする）は、1 個の正解音節のスコア（確率変数を S_{cs} とする）と $N-1$ 音節の誤り音節のスコア（確率変数を S_{is} とする）の最大値で、

$$H(y) = F(Y = \max(S_{cs}, S_{is}^{(1)}, \cdots, S_{is}^{(N-1)}) < y)$$

$$= F_s(y)[G_s(y)]^{N-1}$$

$$h(y) = f_s(y)[G_s(y)]^{N-1} + F_s(y)g_s(y) \cdot (N-1)[G_s(y)]^{N-2}$$

但し、$F_s(y), G_s(y)$ はそれぞれ正解音節スコアと誤り音節スコアの分布関数である。L 音節のスコアの和で得られる未知語のスコア（確率変数を Z とする）の分布関数と確率密度関数は、

$$K(z) = F(Z = Y^{(1)} + Y^{(2)} + \cdots + Y^{(L)} < z)$$

$$= \underbrace{H(z) * H(z) * \cdots * H(z)}_{L\text{個}}$$

$$k(z) = \underbrace{h(z) * h(z) * \cdots * h(z)}_{L\text{個}}$$

但し、"*" は畳み込み演算

となる。

未知語の入力を仮定した時、仮定より、M個の単語のスコアは未知語のスコアと無関係に全て誤り単語のスコア分布 (ここでは $G(x)$ とおく) に従う。従って、未知語入力時の未知語検出率 P_uは、未知語検出のしきい値をTとした時、

$$P_u = \int_{-\infty}^{\infty} k(x)[G(x-T)]^M dx$$

となる。

一方、既知語の入力を仮定した時は、M個のうち一つの単語が正解単語のスコア分布 (確率密度関数 $f(x)$ とする) に従う。ここで、正解単語のスコア分布が未知語スコアと相関が無い (つまり、音節のスコア分布を使わない) と仮定すれば、単語の認識率 P_cは、

$$P_c = \int_{-\infty}^{\infty} f(x) \cdot K(x+T) \cdot [G(x)]^{M-1} dx$$

となる。

C.2 東北大–松下単語音声データベースによる単語認識実験結果

5.3節のリジェクション性能の評価実験に用いた単語音声データベースに対する全単語登録での認識実験結果を示す。認識には、音節単位の HMM を用いた Viterbi アルゴリズムによる連続音声認識法を使用した。評価用の音声資料としては、東北大–松下単語音声データベース中の 212 単語集合の男性話者 15 名 (CD-ROM, Vol.1) のデータを用いた。

音響モデルとして用いた音節単位の HMM は、4.4節で述べた不特定話者モデルである (学習データは、ATR 研究用音声データベース中の男性 6 名による 503 文の連続音声データ、男性 10 名による音韻バランス 216 単語の音声データ、及び、日本音響学会

研究用連続音声データベース中の男性 30 名による計 4500 文の音声データ、の 3 種類からなる)。学習用と評価用のデータの平均的な発声速度は、各データの一部を抜き出して調べたところ、文音声である学習用データでは約 3.7 音節／秒、単語音声である評価用データでは約 6.4 音節／秒で、約 1.7 倍の違いがあったが、継続時間分布等の補正はしなかった。

認識実験では、HMM は回帰係数パラメータなし (MEL) &あり (RGC) の両方を用いた。各話者ごとの 212 単語の認識率を表 C.1に示す。

表 C.1: 東北大・松下データベース 212 単語の認識結果

話者	sp106	sp210	sp301	sp337	sp904	sp907	sp910	sp104
単語数	212	212	212	212	207	211	195	212
MEL	76.9	99.1	97.6	98.6	99.0	96.2	95.9	96.7
RGC	84.4	98.1	100.0	100.0	98.1	97.6	96.4	98.6
話者	sp208	sp226	sp307	sp338	sp905	sp908	sp911	–
単語数	211	212	207	212	210	212	207	–
MEL	98.1	97.2	97.1	91.5	98.1	98.6	98.6	–
RGC	96.2	98.1	97.1	92.5	99.0	99.1	97.6	–

C.3 評価用 115 文リスト (富士山観光&宿泊施設案内タスク)

5.4節の文音声認識におけるリジェクション性能の評価実験に用いた評価用音声データの発話文リストを示す。評価用のタスクは、富士山周辺の観光と宿泊施設の案内に関するもので、4章の評価実験での観光案内タスク (付録 B) と 6章の評価実験での宿泊施設案内タスク (付録 D) を統合して新たに設定したものである。

文の末尾の括弧内の数字は、その文に含まれる言語現象の種類の他、システムの辞書に含まれない未知語を含むか、システムの文法で受理できない文 (文法外) であるかを示している。各数字の意味を以下に示す。

1. 間投詞

2. 助詞落ち

3. 言い直し

4. 倒置

5. 未知語

6. 文法外

1 どんな観光地があるんですかね、河口湖って。(4)
2 富士急ハイランドって何ですか。
3 富士急ハイランドの、えー、入場料はいくらですか。(1)
4 山中湖の宿泊施設にはどんなところがありますか。
5 ホテル、宿泊したいんですが。(2)
6 グラ…、山中ホテルの料金はいくらですか。(3・5)
7 富士博物館ってどこにありますか。
8 富士博物館には、えーと、何がありますか。(1)
9 富士博物館の入館料はいくらですか。(5)
10 河口湖にどんな、りょ…、民宿がありますか。(3)
11 民宿に食事は付いているんでしょうか。
12 やま…、山中湖にあるんですか、鳴沢の氷穴は。(3・4)
13 山中湖に何、ありますか。(2)
14 どこで宿泊できますか。
15 西湖で宿泊したいんですが。
16 いくらぐらいのホテルが、ある…、ありますか。(3)
17 朝食はありますか。
18 サイクリングしたいんですが。
19 河口湖で(他には)何ができますか。
20 テニスのできるホテルは、えー、河口湖にありますか。(1)
21 河口湖ホテルに温泉はありますか。
22 いくらですか、料金は。(4)
23 富士急ハイランド、そのー、遊園地ですよね。(1・2)
24 富士急ハイランドにはどんなアトラクションがありますか。
25 ＺＯＬＡってなんなんですか。
26 富士急ハイランドでスケートできますよね。(2)
27 富士急ハイランドには、ホル…、ホテルがありますか。(3)
28 ハイランドホステルはいくらかかりますか。
29 富士山にはどんな観光地がありますか。
30 富士…、富岳の風穴って洞窟ですか。(3)
31 (他には)どんな洞窟が富士山にあるんですか。
32 鳴沢の氷穴、どこにありますか。(2)
33 泊まるところはどのあたりが多いですかね。(5)
34 どんなホテルが山中湖のあたりにありますか。
35 山中湖ホテル、そのー、泊ま…、宿泊したいんですが。(1・2・3)
36 ホテルはどこにありますか。
37 西湖の近くの、あの、ホテルを考えているんですが。(1)
38 ホテルに宿泊すると宿泊の費用はいくらかかりますか。(6)

39 ついていますか、ホテル西湖に食事は。(4)
40 ホテルに食事はつかないんですか。
41 旅館に夕…、食事つきますか。(2・3)
42 旅館の料金はいくらぐらいですか。
43 テニスを河口湖でしたいんですが。
44 河口湖ホテルには専用テニスコートあるんですか。(2)
45 サイクリングもしたいんですが。(6)
46 へー、サイクリングできるんですね、河…、山中湖で。(1・2・3・4)
47 あと、山中湖で泊まれますか。(1)
48 どんなペンションがありますか。
49 ペンションマリエに食事はありますか。(5)
50 どんなスポーツができますか、西湖で。(4)
51 スケートをしたいんですが。
52 富士急ハイランドって、ど…、どこですか。(3)
53 入場料はいくらですか。
54 泊まるところはどの湖がいいですかね。
55 河口湖の近くに、えー、どんな宿泊施設がありますか。(1)
56 どんなペンションとかがありますか。
57 温泉はありますか、近くに。(4)
58 温泉の料金はいくらかかりますか。
59 どこでキャンプができますか。
60 河口湖にどんなキャンプ場ありますか。(2)
61 食事するところはありますか。
62 テニスできますか、近くで。(2・4)
63 河口湖に、その、遊覧用船はありますか。(1・5)
64 遊覧船に料金はかかりますか。
65 山中湖のあたりに宿泊しようと思っています。
66 ペンションはいくらぐらいかかりますか。
67 んー、５０００円のペンションはありますか。(1)
68 ペンションクレヨンにはペンション自慢の食事が付きますか。(1・6)
69 いくらかかりますか、食事の料金は。(4)
70 富士周辺に遊園地はありますか。(5)
71 どんな遊園地が、ある…、ありますか。(3)
72 富士急ハイランドではどんなジェットコースターがありますか。
73 どこで、えー、乗れるんですか、遊覧船って。(1・4)
74 河口湖で何ができますか。
75 スケート、河口湖ではできないんですか。(2)
76 どこでスケートができるんですか。
77 山中湖のホテルに宿泊しようと思っているんですが。
78 いくらかかると思いますか、お金は。(4)
79 どんなスポーツ、富士山界隈でできますか。(2・5)
80 どの湖にキャンプ場がありますか。
81 河口湖の、まわ…、周辺に富士博物館がありますよね。(3)
82 周辺に宿泊施設はありますか、富士博物館の。(4)
83 ホテルに泊まりたいんですが。

８４　ホテルはありますか、温泉付きの。（６）
８５　どの辺に宿泊施設ありますか。（２）
８６　いくらぐらいになりますか、河口湖の旅館は、えーと、宿泊の料金。（１・２・４）
８７　ホテルではいくらぐらいに…、ですか。（３）
８８　じゃ、ホテルがいいです。（１）
８９　これらのホテルに食事はありますか。
９０　宿泊しても食事の料金はかかりますよね。（６）
９１　河口湖って富士山の観光地ですよね。
９２　富士博物館どこにあるんですか。（２）
９３　鳴沢の氷穴ありましたよね、えー、河口湖に。（１・２・４）
９４　鳴沢の氷穴はど…、どこにありますか。（３）
９５　そこには展望台がありますよね。
９６　どこでできますか、水上スキーは。（４）
９７　どんな旅館がその近くにあるんですか。
９８　えーと、安い旅館はありますか。（１）
９９　６０００円です。
１００　朝食が付いてですか。
１０１　富士山周辺の観光したいんですが。（２）
１０２　えー、そう…、そうですね。（１・３）
１０３　いいえ、精進湖の辺りで泊まりたいんですが。（５）
１０４　シングルです。
１０５　お願いします。
１０６　温泉は富士山にありますか。
１０７　温泉のあるホテ…、旅館に泊まりたいんですが。（３、６）
１０８　安い方が良いです。
１０９　露天風呂があるところかな。（６）
１１０　どこにありますか、その旅館は。（４）
１１１　どんな氷穴が富士山にありますか。
１１２　鳴沢の氷穴はつめ…、涼しいですか。（３）
１１３　河口湖の近くですか。
１１４　河口湖の近くに、その、富士山があるんですよね。（１）
１１５　有難う。

付録 D.

「宿泊施設案内」タスクによる評価実験

6章の自然な発話のための認識方式の比較・評価実験でのタスクの文法と評価用音声データの発話文リストを示す。

D.1 Island-Driven 法のシステムの文法（文節単位の意味的文法）

文法の構文レベルの記述は、6.3.2節で述べるように２種類の書き換え規則からなる。(a) の文レベルの規則では、

SENT　A　B　C

と記述される場合、A, B, C の３つの非終端記号に対応する文節の全ての可能な並びを仮定した文を受理する。従って、この規則の例では、以下の文脈自由文法の規則と等価な構文的制約を持つ。

SENT → A　B　C
SENT → A　C　B
SENT → B　A　C
SENT → B　C　A
SENT → C　A　B
SENT → C　B　A

一方、(b) の文節間レベルの規則では、“A　B　C” という記法は、“A → B　C” という文脈自由文法の書き換え規則と等価な表現である。

(a) 文レベルの規則

```
SENT VERB1 Q-OBJ1 ACCOMM
SENT VERB1 Q-OBJ2
SENT VERB1 Q-OBJ2 ACCOMM
SENT VERB1 AT-LOC
SENT VERB1 ACCOMM
SENT VERB1 ACCOMM AT-LOC
SENT VERB2 ACCOMM OPTION
SENT VERB2 OPTION
SENT VERB2 ACCOMM
SENT VERB2 AT-LOC ACCOMM
SENT VERB2 Q-OBJ1
SENT VERB2 Q-OBJ1 AT-LOC
SENT VERB2 Q-OBJ1 ACCOMM
SENT VERB2 Q-OBJ2
SENT VERB2 Q-OBJ2 ACCOMM
SENT VERB2 Q-OBJ2 AT-LOC
SENT VERB2 Q-OBJ3 ACCOMM
SENT VERB3 Q-OBJ3
SENT VERB3 Q-OBJ3 RATE
SENT VERB3 Q-OBJ3 ACCOMM
SENT VERB3 Q-OBJ3 ACCOMM RATE
SENT VERB3 Q-OBJ3 RATE VERB1 ACCOMM
SENT VERB3 Q-OBJ3 ACCOMM RATE
SENT VERB3 Q-OBJ3 RATE VERB1 ACCOMM
```

(b) 文節間レベルの規則

```
VERB1 stay
VERB1 want
VERB1 a-lot
VERB1 THINK
VERB2 isthere
VERB2 have
VERB3 costs
VERB3 costs think
THINK think
THINK stay think
THINK want think
Q-OBJ1 what
Q-OBJ2 where
Q-OBJ3 how-much
AT-LOC PLACE1
AT-LOC PLACE2
AT-LOC near
PLACE1 Place1
PLACE1 Place1 near
Place1 place1
Place1 place1 place1
Place1 place1 not place1
PLACE2 place2
PLACE2 place2 near
PLACE2 where near
ACCOMM Accomm
ACCOMM rate2 Accomm
ACCOMM near Accomm
Accomm accomm
Accomm accomm accomm
Accomm accomm not accomm
RATE rate1
RATE Accomm rate1
RATE OPTION rate1
OPTION option
OPTION option option
INT int
INT not
```


D.2 システムの文法（文脈自由文法）

Left-to-Right 型解析法によるシステム（6.3.3節参照）で用いる文脈自由文法による構文規則を以下に示す（記法は付録 B.1と同じ）。前節の文法と構文的な制約は等価である。One-Pass 法に基づくシステム（6.4節参照）で用いる文法も基本的にこの文法を元にしているが、文節間における言い直しや間投詞の挿入に対して未知語処理を適用するために、書き換え規則の全ての文節の非終端記号の間に、未知語を仮定した非終端記号（空終端記号または “未知語” を導出）を挿入して用いる。

```
SSSS  SENT

SENT VERB1  Q-OBJ1 ACCOMM
SENT VERB1  ACCOMM Q-OBJ1
SENT Q-OBJ1 ACCOMM VERB1
SENT Q-OBJ1 VERB1  ACCOMM
SENT ACCOMM Q-OBJ1 VERB1
SENT ACCOMM VERB1  Q-OBJ1

SENT VERB1 Q-OBJ2
SENT Q-OBJ2 VERB1

SENT VERB1  Q-OBJ2 ACCOMM
SENT VERB1  ACCOMM Q-OBJ2
SENT Q-OBJ2 VERB1  ACCOMM
SENT Q-OBJ2 ACCOMM VERB1
SENT ACCOMM VERB1  Q-OBJ2
SENT ACCOMM Q-OBJ2 VERB1

SENT VERB1 AT-LOC
SENT AT-LOC VERB1

SENT VERB1 ACCOMM
SENT ACCOMM VERB1

SENT VERB1 ACCOMM AT-LOC
SENT VERB1 AT-LOC ACCOMM
SENT AT-LOC ACCOMM VERB1
SENT AT-LOC VERB1 ACCOMM
SENT ACCOMM VERB1 AT-LOC
SENT ACCOMM AT-LOC VERB1

SENT VERB2 ACCOMM OPTION
SENT VERB2 OPTION ACCOMM
SENT OPTION ACCOMM VERB2
SENT OPTION VERB2 ACCOMM
SENT ACCOMM VERB2 OPTION
SENT ACCOMM OPTION VERB2

SENT VERB2 OPTION
SENT OPTION VERB2

SENT VERB2 ACCOMM
SENT ACCOMM VERB2

SENT VERB2 AT-LOC ACCOMM
SENT VERB2 ACCOMM AT-LOC
SENT ACCOMM AT-LOC VERB2
SENT ACCOMM VERB2 AT-LOC
SENT AT-LOC VERB2 ACCOMM
SENT AT-LOC ACCOMM VERB2

SENT VERB2 Q-OBJ1
SENT Q-OBJ1 VERB2

SENT VERB2 Q-OBJ1 AT-LOC
SENT VERB2 AT-LOC Q-OBJ1
SENT AT-LOC Q-OBJ1 VERB2
SENT AT-LOC VERB2 Q-OBJ1
SENT Q-OBJ1 VERB2 AT-LOC
SENT Q-OBJ1 AT-LOC VERB2

SENT VERB2 Q-OBJ1 ACCOMM
SENT VERB2 ACCOMM Q-OBJ1
SENT ACCOMM Q-OBJ1 VERB2
SENT ACCOMM VERB2 Q-OBJ1
SENT Q-OBJ1 VERB2 ACCOMM
SENT Q-OBJ1 ACCOMM VERB2

SENT VERB2 Q-OBJ2
SENT Q-OBJ2 VERB2

SENT VERB2 Q-OBJ2 ACCOMM
SENT VERB2 ACCOMM Q-OBJ2
SENT ACCOMM Q-OBJ2 VERB2
SENT ACCOMM VERB2 Q-OBJ2
SENT Q-OBJ2 ACCOMM VERB2
SENT Q-OBJ2 VERB2 ACCOMM

SENT VERB2 Q-OBJ2 AT-LOC
SENT VERB2 AT-LOC Q-OBJ2
SENT AT-LOC Q-OBJ2 VERB2
SENT AT-LOC VERB2 Q-OBJ2
SENT Q-OBJ2 AT-LOC VERB2
SENT Q-OBJ2 VERB2 AT-LOC

SENT VERB2 Q-OBJ3 ACCOMM
SENT VERB2 ACCOMM Q-OBJ3
```

```
SENT ACCOMM Q-OBJ3 VERB2
SENT ACCOMM VERB2 Q-OBJ3
SENT Q-OBJ3 ACCOMM VERB2
SENT Q-OBJ3 VERB2 ACCOMM

SENT VERB3 Q-OBJ3
SENT Q-OBJ3 VERB3

SENT VERB3 Q-OBJ3 RATE
SENT VERB3 RATE Q-OBJ3
SENT RATE Q-OBJ3 VERB3
SENT RATE VERB3 Q-OBJ3
SENT Q-OBJ3 RATE VERB3
SENT Q-OBJ3 VERB3 RATE

SENT VERB3 Q-OBJ3 ACCOMM
SENT VERB3 ACCOMM Q-OBJ3
SENT Q-OBJ3 ACCOMM VERB3
SENT Q-OBJ3 VERB3 ACCOMM
SENT ACCOMM VERB3 Q-OBJ3
SENT ACCOMM Q-OBJ3 VERB3

SENT VERB3 Q-OBJ3 ACCOMM RATE
SENT VERB3 Q-OBJ3 RATE ACCOMM
SENT VERB3 ACCOMM RATE Q-OBJ3
SENT VERB3 ACCOMM Q-OBJ3 RATE
SENT VERB3 RATE Q-OBJ3 ACCOMM
SENT VERB3 RATE ACCOMM Q-OBJ3
SENT Q-OBJ3 VERB3 ACCOMM RATE
SENT Q-OBJ3 VERB3 RATE ACCOMM
SENT Q-OBJ3 ACCOMM RATE VERB3
SENT Q-OBJ3 ACCOMM VERB3 RATE
SENT Q-OBJ3 RATE VERB3 ACCOMM
SENT Q-OBJ3 RATE ACCOMM VERB3
SENT ACCOMM Q-OBJ3 VERB3 RATE
SENT ACCOMM Q-OBJ3 RATE VERB3
SENT ACCOMM VERB3 RATE Q-OBJ3
SENT ACCOMM VERB3 Q-OBJ3 RATE
SENT ACCOMM RATE Q-OBJ3 VERB3
SENT ACCOMM RATE VERB3 Q-OBJ3
SENT RATE Q-OBJ3 ACCOMM VERB3
SENT RATE Q-OBJ3 VERB3 ACCOMM
SENT RATE ACCOMM VERB3 Q-OBJ3
SENT RATE ACCOMM Q-OBJ3 VERB3
SENT RATE VERB3 Q-OBJ3 ACCOMM
SENT RATE VERB3 ACCOMM Q-OBJ3

SENT VERB1 VERB3 Q-OBJ3 ACCOMM RATE
SENT VERB1 VERB3 Q-OBJ3 RATE ACCOMM
SENT VERB1 VERB3 ACCOMM RATE Q-OBJ3
SENT VERB1 VERB3 ACCOMM Q-OBJ3 RATE
SENT VERB1 VERB3 RATE Q-OBJ3 ACCOMM
SENT VERB1 VERB3 RATE ACCOMM Q-OBJ3
SENT VERB1 Q-OBJ3 VERB3 ACCOMM RATE
SENT VERB1 Q-OBJ3 VERB3 RATE ACCOMM
SENT VERB1 Q-OBJ3 ACCOMM RATE VERB3
SENT VERB1 Q-OBJ3 ACCOMM VERB3 RATE
```

```
SENT VERB1 Q-OBJ3 RATE VERB3 ACCOMM
SENT VERB1 Q-OBJ3 RATE ACCOMM VERB3
SENT VERB1 ACCOMM Q-OBJ3 VERB3 RATE
SENT VERB1 ACCOMM Q-OBJ3 RATE VERB3
SENT VERB1 ACCOMM VERB3 RATE Q-OBJ3
SENT VERB1 ACCOMM VERB3 Q-OBJ3 RATE
SENT VERB1 ACCOMM RATE Q-OBJ3 VERB3
SENT VERB1 ACCOMM RATE VERB3 Q-OBJ3
SENT VERB1 RATE Q-OBJ3 ACCOMM VERB3
SENT VERB1 RATE Q-OBJ3 VERB3 ACCOMM
SENT VERB1 RATE ACCOMM VERB3 Q-OBJ3
SENT VERB1 RATE ACCOMM Q-OBJ3 VERB3
SENT VERB1 RATE VERB3 Q-OBJ3 ACCOMM
SENT VERB1 RATE VERB3 ACCOMM Q-OBJ3
SENT VERB3 VERB1 Q-OBJ3 ACCOMM RATE
SENT VERB3 VERB1 Q-OBJ3 RATE ACCOMM
SENT VERB3 VERB1 ACCOMM RATE Q-OBJ3
SENT VERB3 VERB1 ACCOMM Q-OBJ3 RATE
SENT VERB3 VERB1 RATE Q-OBJ3 ACCOMM
SENT VERB3 VERB1 RATE ACCOMM Q-OBJ3
SENT VERB3 Q-OBJ3 VERB1 ACCOMM RATE
SENT VERB3 Q-OBJ3 VERB1 RATE ACCOMM
SENT VERB3 Q-OBJ3 ACCOMM RATE VERB1
SENT VERB3 Q-OBJ3 ACCOMM VERB1 RATE
SENT VERB3 Q-OBJ3 RATE VERB1 ACCOMM
SENT VERB3 Q-OBJ3 RATE ACCOMM VERB1
SENT VERB3 ACCOMM Q-OBJ3 VERB1 RATE
SENT VERB3 ACCOMM Q-OBJ3 RATE VERB1
SENT VERB3 ACCOMM VERB1 RATE Q-OBJ3
SENT VERB3 ACCOMM VERB1 Q-OBJ3 RATE
SENT VERB3 ACCOMM RATE Q-OBJ3 VERB1
SENT VERB3 ACCOMM RATE VERB1 Q-OBJ3
SENT VERB3 RATE Q-OBJ3 ACCOMM VERB1
SENT VERB3 RATE Q-OBJ3 VERB1 ACCOMM
SENT VERB3 RATE ACCOMM VERB1 Q-OBJ3
SENT VERB3 RATE ACCOMM Q-OBJ3 VERB1
SENT VERB3 RATE VERB1 Q-OBJ3 ACCOMM
SENT VERB3 RATE VERB1 ACCOMM Q-OBJ3
SENT Q-OBJ3 VERB3 VERB1 ACCOMM RATE
SENT Q-OBJ3 VERB3 VERB1 RATE ACCOMM
SENT Q-OBJ3 VERB3 ACCOMM RATE VERB1
SENT Q-OBJ3 VERB3 ACCOMM VERB1 RATE
SENT Q-OBJ3 VERB3 RATE VERB1 ACCOMM
SENT Q-OBJ3 VERB3 RATE ACCOMM VERB1
SENT Q-OBJ3 VERB1 VERB3 ACCOMM RATE
SENT Q-OBJ3 VERB1 VERB3 RATE ACCOMM
SENT Q-OBJ3 VERB1 ACCOMM RATE VERB3
SENT Q-OBJ3 VERB1 ACCOMM VERB3 RATE
SENT Q-OBJ3 VERB1 RATE VERB3 ACCOMM
SENT Q-OBJ3 VERB1 RATE ACCOMM VERB3
SENT Q-OBJ3 ACCOMM VERB1 VERB3 RATE
SENT Q-OBJ3 ACCOMM VERB1 RATE VERB3
SENT Q-OBJ3 ACCOMM VERB3 RATE VERB1
SENT Q-OBJ3 ACCOMM VERB3 VERB1 RATE
SENT Q-OBJ3 ACCOMM RATE VERB1 VERB3
SENT Q-OBJ3 ACCOMM RATE VERB3 VERB1
SENT Q-OBJ3 RATE VERB1 ACCOMM VERB3
```



```
SENT Q-OBJ3 RATE VERB1 VERB3 ACCOMM
SENT Q-OBJ3 RATE ACCOMM VERB3 VERB1
SENT Q-OBJ3 RATE ACCOMM VERB1 VERB3
SENT Q-OBJ3 RATE VERB3 VERB1 ACCOMM
SENT Q-OBJ3 RATE VERB3 ACCOMM VERB1
SENT ACCOMM VERB3 Q-OBJ3 VERB1 RATE
SENT ACCOMM VERB3 Q-OBJ3 RATE VERB1
SENT ACCOMM VERB3 VERB1 RATE Q-OBJ3
SENT ACCOMM VERB3 VERB1 Q-OBJ3 RATE
SENT ACCOMM VERB3 RATE Q-OBJ3 VERB1
SENT ACCOMM VERB3 RATE VERB1 Q-OBJ3
SENT ACCOMM Q-OBJ3 VERB3 VERB1 RATE
SENT ACCOMM Q-OBJ3 VERB3 RATE VERB1
SENT ACCOMM Q-OBJ3 VERB1 RATE VERB3
SENT ACCOMM Q-OBJ3 VERB1 VERB3 RATE
SENT ACCOMM Q-OBJ3 RATE VERB3 VERB1
SENT ACCOMM Q-OBJ3 RATE VERB1 VERB3
SENT ACCOMM VERB1 Q-OBJ3 VERB3 RATE
SENT ACCOMM VERB1 Q-OBJ3 RATE VERB3
SENT ACCOMM VERB1 VERB3 RATE Q-OBJ3
SENT ACCOMM VERB1 VERB3 Q-OBJ3 RATE
SENT ACCOMM VERB1 RATE Q-OBJ3 VERB3
SENT ACCOMM VERB1 RATE VERB3 Q-OBJ3
SENT ACCOMM RATE Q-OBJ3 VERB1 VERB3
SENT ACCOMM RATE Q-OBJ3 VERB3 VERB1
SENT ACCOMM RATE VERB1 VERB3 Q-OBJ3
SENT ACCOMM RATE VERB1 Q-OBJ3 VERB3
SENT ACCOMM RATE VERB3 Q-OBJ3 VERB1
SENT ACCOMM RATE VERB3 VERB1 Q-OBJ3
SENT RATE VERB3 Q-OBJ3 ACCOMM VERB1
SENT RATE VERB3 Q-OBJ3 VERB1 ACCOMM
SENT RATE VERB3 ACCOMM VERB1 Q-OBJ3
SENT RATE VERB3 ACCOMM Q-OBJ3 VERB1
SENT RATE VERB3 VERB1 Q-OBJ3 ACCOMM
SENT RATE VERB3 VERB1 ACCOMM Q-OBJ3
SENT RATE Q-OBJ3 VERB3 ACCOMM VERB1
SENT RATE Q-OBJ3 VERB3 VERB1 ACCOMM
SENT RATE Q-OBJ3 ACCOMM VERB1 VERB3
SENT RATE Q-OBJ3 ACCOMM VERB3 VERB1
SENT RATE Q-OBJ3 VERB1 VERB3 ACCOMM
SENT RATE Q-OBJ3 VERB1 ACCOMM VERB3
SENT RATE ACCOMM Q-OBJ3 VERB3 VERB1
SENT RATE ACCOMM Q-OBJ3 VERB1 VERB3
SENT RATE ACCOMM VERB3 VERB1 Q-OBJ3
SENT RATE ACCOMM VERB3 Q-OBJ3 VERB1
SENT RATE ACCOMM VERB1 Q-OBJ3 VERB3
SENT RATE ACCOMM VERB1 VERB3 Q-OBJ3
```

```
SENT RATE VERB1 Q-OBJ3 ACCOMM VERB3
SENT RATE VERB1 Q-OBJ3 VERB3 ACCOMM
SENT RATE VERB1 ACCOMM VERB3 Q-OBJ3
SENT RATE VERB1 ACCOMM Q-OBJ3 VERB3
SENT RATE VERB1 VERB3 Q-OBJ3 ACCOMM
SENT RATE VERB1 VERB3 ACCOMM Q-OBJ3


VERB1 stay
VERB1 want
VERB1 a-lot
VERB1 THINK
VERB2 isthere
VERB2 have
VERB3 costs
VERB3 costs think
THINK think
THINK stay think
THINK want think
Q-OBJ1 what
Q-OBJ2 where
Q-OBJ3 how-much
AT-LOC PLACE1
AT-LOC PLACE2
AT-LOC near
PLACE1 PLACE-1
PLACE1 PLACE-1 near
PLACE-1 place1
PLACE-1 place1 place1
PLACE-1 place1 not place1
PLACE2 place2
PLACE2 place2 near
PLACE2 where near
ACCOMM ACCOMMS
ACCOMM rate2 ACCOMMS
ACCOMM near ACCOMMS
ACCOMMS accomm
ACCOMMS accomm accomm
ACCOMMS accomm not accomm
RATE rate1
RATE ACCOMMS rate1
RATE OPTION rate1
OPTION option
OPTION option option
INT int
INT not
```

D.3 評価用 70 文リスト

＜施設＞
No. 1 宿泊施設には、どんなところがありますか。
No. 2 その辺で宿泊しようと思っています。
No. 3 どんなペンションとかがありますか。
No. 4 その周辺に、どのような民宿がありますか。
No. 5 どんなホテルが、河口湖のあたりにありますか。
No. 6 西湖で、宿泊したいんですが。
No. 7 ホテルに宿泊しようと思っています。
No. 8 本栖湖の近くにホテルかペンションがありますか。
No. 9 精進湖の近くに、どんな宿泊施設がありますか。
No.10 どんな旅館がその近くにあるんですか。

＜料金＞
No.11 だいたい、いくらぐらいになりますか。
No.12 お金は、いくらかかると思いますか。
No.13 ペンションは、いくらかかりますか。
No.14 ホテルに宿泊すると、宿泊の費用はいくらかかりますか。
No.15 旅館は、宿泊の料金がいくらぐらいになりますか。

＜場所＞
No.16 泊まるところは、どの湖が多いですかね。
No.17 ホテルはどこにありますか。
No.18 どこで宿泊できますか。

＜宿泊内容＞
No.19 食事はつきませんか。
No.20 温泉の料金は、いくらかかりますか。

＜間投詞を含む文＞
No.21 その周辺に、【えーと】、どのような民宿がありますか。
No.22 西湖で、【え】宿泊したいんですが。
No.23 【まーー】ホテルで宿泊しようと思っています。
No.24 【あの】、お金は、いくらかかると思いますか。
No.25 その周辺に、【えーと】、どのような民宿が【あーー】ありますか。

＜施設＞
No.26 河口湖か山中湖のあたりに宿泊しようと思っています。
No.27 どんな泊まるところがありますか。
No.28 山中湖のホテルに宿泊しようと思っています。
No.29 西湖の近くのホテルを考えているんですが。
No.30 どんな旅館があるんですか。その近くには。

＜料金＞

No.31 そのペンションは、いくらかかりますか。
No.32 旅館は、宿泊料金がいくらぐらいになりますか。
No.33 いくらぐらいのホテルがありますか。
No.34 食事の料金は、いくらかかりますか。
No.35 その辺の旅館はいくらかかりますか。
No.36 いくらかかりますか。ペンションは。

＜場所＞
No.37 泊まるところは、どのあたりが多いですかね。
No.38 泊まるところは、どの湖のあたりに多いですかね。
No.39 西湖のどこにありますか。
No.40 どの辺に宿泊施設がありますか。
No.41 どこが多いですか。泊まるところは。

＜宿泊内容＞
No.42 民宿に、食事とかはついているんでしょうか。
No.43 温泉とかはついているんでしょうか。
No.44 食事はありますか。
No.45 つきますか。食事とかは。

＜間投詞を含む文＞
No.46 温泉の料金は、いくら【あーー】かかりますか。
No.47 泊まるところは、【じゃあ】、どの湖が多いですかね。
No.48 その辺で宿泊【えー】しようと思っています。
No.49 【えーと】 【あの】 どんなペンションがありますか。
No.50 【えーと】、どのような民宿がありますか。その周辺に。

＜助詞落ち＞
No.51 どんなところに、宿泊施設ありますか。
No.52 食事つきませんか。
No.53 その近くには、どんな旅館あるんですか。
No.54 どこが多い。泊まるところ。
No.55 つきますか。食事。
No.56 お金、【えーー】いくらかかりますか。
No.57 泊まるところ、【じゃあ】どの湖多いですかね。
No.58 その周辺、【えーと】、どのような民宿が【あーー】ありますか。
No.59 そのペンション【あーー】、いくらかかりますか。
No.60 【えーと】、どのような民宿ありますか。その周辺に。

＜言い直し＞
No.61 どんな（旅館、いや、）ホテルが、河口湖のあたりにありますか。
No.62 （にしー、さ、）西湖のどこにありますか。
No.63 民宿に、（食‥、）食事とかはついているんでしょうか。
No.64 温泉（に、）は、ついているんでしょうか。
No.65 （おん‥、）温泉の料金は、いくらかかりますか。

No.66 どんな（り、）【えーと】旅館あるんですか。その近くには。
No.67 泊まるところは、（多<oo>‥）【えーと】どの湖に多いですかね。
No.68 旅館は、（おか）【あーー】宿泊料金がいくらぐらいになりますか。
No.69 （やまな、）【えーと】西湖のどこにありますか。
No.70 そのペンションは、いくらに（なり‥）【あのー】かかりますか。

発表論文

1. 学会論文誌

[1] Jun-ichi Takami, Atsuhiko Kai, and Shigeki Sagayama: "A pairwise discriminant approach using artificial neural networks for continuous speech recognition", *The Journal of the Acoustical Society of Japan (E)*, Vol.13, No.6, pp.411–418, 1992.

[2] 中川 聖一, 甲斐 充彦: "ワードスポッティング法を用いた文脈自由文法制御フレーム同期型 HMM 連続音声認識法", 電子情報通信学会論文誌, Vol.J76-D-II, No.7, pp.1329–1336, 1993.

[3] 中川 聖一, 甲斐 充彦: "文脈自由文法制御による One Pass 型 HMM 連続音声認識法", 電子情報通信学会論文誌, Vol.J76-D-II, No.7, pp.1337–1345, 1993.

[3]' Seiichi Nakagawa and Atsuhiko Kai: "A context-free grammar-driven, One-Pass HMM-based continuous speech recognition method", *Systems and Computers in Japan*, Vol.25, No.4, pp.92–102, 1994.
（上記論文の英訳版）

[4] Atsuhiko Kai and Seiichi Nakagawa: "Relationship among recognition rate, rejection rate and false alarm rate in a spoken word recognition system", *IEICE Trans. on Information and Systems*, Vol.E-78-D, No.6, pp.698–704, 1995.

2. 国際会議発表論文

[1] Atsuhiko Kai and Seiichi Nakagawa: "A frame-synchronous continuous speech recognition algorithm using a top-down parsing of context-free grammar," *Proc. of International Conference on Spoken Language Processing*, Alberta, Canada, pp.257–260, 1992.

[2] Atsuhiko Kai and Seiichi Nakagawa: "Evaluation of unknown word processing in a spoken word recognition system," *Proc. of International Conference on Spoken Language Processing*, Yokohama, Japan, pp.2151–2154, 1994.

[3] Atsuhiko Kai and Seiichi Nakagawa: "Investigation on unknown word processing and strategies for spontaneous speech understanding," *Proc. of EUROSPEECH'95*, Madrid, Spain, pp.2095–2098, 1995.

3. 学会・研究会発表論文

[1] 甲斐充彦, 中川聖一：文脈自由文法の構文解析法を用いたHMM連続音声認識システム, 電子情報通信学会, 第2種研究会, SPREC-91-1, pp.55–58 (1991.7).

[2] 甲斐充彦, 中川聖一：文脈自由文法制御フレーム同期型HMM連続音声認識アルゴリズム, 日本音響学会講論集, 1-5-15 (1991.10).

[3] 甲斐充彦, 中川聖一：二つの文脈自由文法制御フレーム同期型連続音声認識アルゴリズム, 情報処理学会第43回全国大会講演論文集(2), 5V-7, pp.545–546 (1991.10).

[4] 甲斐充彦, 中川聖一：文脈自由文法制御フレーム同期型HMM連続音声認識アルゴリズムとその高速化, 電子情報通信学会技術報告, Vol.91, SP91-91 (1991.12).

[5] 甲斐充彦, 中川聖一：文脈自由文法制御と連続ワードスポッティング法によるHMM連続音声認識, 日本音響学会講論集, 1-P-4 (1992.3).

[6] 甲斐充彦, 中川聖一：ワードスポッティング法を用いたHMM連続音声認識についての検討, 電子情報通信学会技術報告, SP92-10 (1992.5).

[7] 小林 聡, 甲斐充彦, 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, 電子情報通信学会, 第2種研究会, SPREC-92-3, pp.21–25 (1993.2).

[8] 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 (1993.6).

[9] 甲斐充彦, 中川聖一：連続音声認識システムSPOJUS-SYNOにおける間投詞・未知語処理の検討, 日本音響学会講論集, 3-7-5 (1993.10).

[10] 甲斐充彦, 中川聖一：音声認識システムにおける未知語処理の評価, 日本音響学会講論集, 2-7-1 (1994.3).

[11] 甲斐充彦, 中川聖一：未知語検出率のシミュレーションと孤立単語及び文音声認識による評価, 電子情報通信学会技術報告, SP94-26 (1994.6).

[12] 甲斐充彦, 間宮康之, 中川聖一：自然発話の認識・理解のための解析・照合手法の比較, 情報処理学会研究報告, 94-SLP-2-12, pp.83–90 (1994.7).

[13] 山本幹雄, 肥田野勝, 伊藤敏彦, 甲斐充彦, 中川聖一：自然発話の意味理解と対話システム, 情報処理学会研究報告, 94-SLP-2-13, pp.91–98 (1994.7).

[14] 鳥居美和子, 甲斐充彦, 中川聖一, 中西宏文：任意語彙の簡易追加登録型単語音声認識法, 日本音響学会講論集, 2-2-1 (1995.9).

[15] 堤真理子, 周 旻, 甲斐充彦, 中川聖一：対話音声認識の言語制約としての文脈自由文法と統計的モデルの比較：日本音響学会講論集, 1-P-7 (1996.3).