SP90－74
NLC90－46

拡張 LR 構文解析法を用いた連続音声認識
Continuous Speech Recognition by the generalized LR Parsing

伊藤克亘† 速水 悟‡ 田中穂積†
ITOU Katunobu HAYAMIZU Satoru TANAKA Hozumi
†東京工業大学工学部
Tokyo Institute of Technology
‡電子技術総合研究所
Electrotechnical Laboratory

あらまし 本稿では、大規模な辞書を用いて音声のパラメータ系列から直接複数の形態素候補を得ることのできる辞書引きアルゴリズムと、辞書引きの結果得られる複数の形態素候補から文節・文などのより大きい単位を構成していく構文解析アルゴリズムについて述べる。辞書引きアルゴリズムは、拡張 LR 構文解析法と One Pass DP アルゴリズムに基づいており、フレーム同期で処理を進めながら、最終的に最適性の高い複数の音韻系列を、語彙数には直接影響を受けない計算量で得ることができる。構文解析アルゴリムも、拡張 LR 構文解析法に基づいているため、辞書引きアルゴリズムと構文解析アルゴリズムの親和性は高い。

Abstract This paper presents two algorithms. One is for matching an acoustic symbol string against entries in the dictionary. The other is for parsing N-best morphoneme strings which are the output of the first algorithm. The first one can be viewed as an extension of the One Pass Dynamic Programming Search Algorithm guided by the dictionary reference which is based on the generalized LR parser. The computational complexity of the algorithm is independent of the number of entries in the dictionary. Because both algorithms are based on generalized LR parsing, it is easy to integrate them.

1 はじめに

音声による自然言語を用いたマンマシンインターフェイスを考えるとき、大語彙で連続音声が扱えることが必須条件となる。

日本語には、語順が自由であるという特徴がある。そのため、単語・形態素のレベルでは、附属語の部分を除けば、埋込文などの可能性を考慮すると次に発話されるものをありえるかありえないかという形で予測するのは非常に困難である。したがって、日本語による自然言語インタフェースでの音声認識方式としては、認識の単位を音韻などのレベルにとり、それらの組合せで形態素や文節などのより大きい単位を構成していく方式 [4] [3] が有望であると考えられる。

しかし、このような方式を実現するためには、音声レベルでは、十分な音韻の認識精度が必要である。この点に関しては近年、HMM (Hidden Markov Model) に基づく音声認識手法の発展により、音韻認識の精度が向上し、これに基づくシステムも SPHYNX [10] をはじめとして数多く開発されている。

一方、自然言語処理のレベルでは、まず、音声のパラメータ系列に対して大規模な辞書を用いた場合に効率のよい辞書引きが行えることが必要であり、次にそれらの結果得られた形態素や単語を効率的に組合せる手法が必要となる。

われわれは、音声を含めた日本語による自然言語インターフェイスを目指すシステム NINJA (Natural language INterface in JApanese) の試作を行っており、本稿ではそのシステムの音声認識部について報告を行う。

NINJA では、音声のパラメータ系列に対してフレーム同期で直接解析を行い、最適な音韻系列 (文の認識結果) を複数個得ようとする方法を採用しており、途中に音韻ラティスや単語ラティスのようなものを介さないのを特徴としている。

本システムでは、音韻モデルには HMM を用いている。音韻を組合せながら、辞書引きを行う部分には One Pass DP アルゴリズム [9] を基本にした経路探索を、拡張 LR 構文解析法を用いた辞書引きアルゴリズム [6] を用いて制御する。辞書引き結果を組合せていく部分には拡張 LR 構文解析法 [13] を用いる。

これらを統合した処理を行う場合には、演算量を削減させるためにも精度向上のためにも、なるべく無駄な再計算を避ける工夫が様々なレベルで必要となる。本稿では、これらを実現するための機構を中心に報告し、最後

に本システムの動作確認のための簡単な実験結果について報告する。

2 システムの構成

システムの構成を図1に示す。システムは大きく分けて構文解析の部分と辞書引きの部分の2つの部分から構成されている。

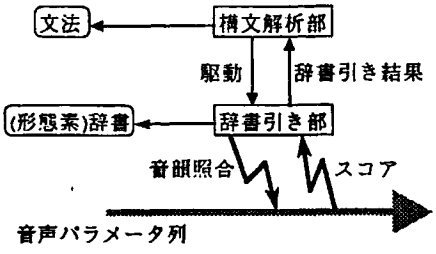

図1：システムの概略図

処理はこの2つの部分が相互に情報を受渡すことによって進められる。

まず、構文解析部が予め用意された文法規則にしたがって、辞書引き処理を駆動させる。辞書引きの部分は、予め用意された辞書を参照して音韻HMMを駆動する。そして、音韻ラティスなどを構成することなく、音声入力のパラメータ系列から、並列に複数の辞書引きをフレーム同期をとりながら行う。また、解析途中で経路の数が爆発的に増大しないように各フレーム毎に枝刈りを行う。

辞書引きが完了すると、辞書引き部は形態素の構文的なカテゴリを構文解析部に渡す。結果を渡された構文解析部は、その形態素を含む経路の適切さを構文などの制約によって判別し、文法規則にしたがって次の辞書引き処理を駆動する。

このような処理を繰返して、解析結果を次々に出力していく。

このように、本システムでは音声認識と同時に自然言語処理も行っている。また、音韻ラティスや単語ラティスなどの中間的な構造を介さないので、効率的かつ高精度な処理を行える。

3 拡張LR構文解析法によるOne Pass DPアルゴリズムの制御

本節では、システムの辞書引き部分に用いるアルゴリズムについて述べる。

3.1 One Pass DPアルゴリズム

One Pass DPアルゴリズム[9]は、有限状態オートマトンによる比較的単純な制御構造を持ち、入力音声のフレームに同期して処理が進むという優れた特徴を持つ。このため、多くの連続単語認識システムで、この手法に基づいた手法が用いられている[12][7][1]。

本システムは、単語ではなく音韻が認識単位であるが、音韻系列を得る場合にも単語系列を得る場合と同様にこの手法を用いることができる。

本システムで、One Pass DP法を制御するのに用いる拡張LR構文解析法に基づく辞書引きアルゴリズムは、有限状態オートマトンと等価な辞書表現を用いるが、大規模な辞書を構築した場合にはその状態数が非常に多くなってしまう。One Pass DPアルゴリズムでは、各フレームで全ての状態のスコアを計算し、その値を保持しなければならないので、計算量が非常に多くなってしまう。

こういった問題に対して、従来の手法では各フレームである一定以上のスコアの状態に対してだけ処理を進めていくことによって制御を行っており、その方法は、最適性をそれほど低下させずに計算量を大幅に削減できることが知られている[11]。

しかし、本システムでは、One Pass DPアルゴリズムを用いた辞書引き結果を組合せて、文節や文などを構成するので、さらにできるだけ計算量を削減することを考える。

ここでは、辞書を有限状態オートマトンで表現した場合の特徴である以下の2点に着目して計算量を削減する手法を考慮する。

1. 辞書が大規模になっても使用される音韻モデル数はある一定数である。
2. 異なる経路に同じ音韻が何度も現れる。

3.2 拡張LR構文解析法を用いた辞書引きアルゴリズム

拡張LR構文解析法を用いた辞書引き[6]を行うための辞書用の規則を次に示す。

```
名詞 -> h o N
名詞 -> k o
名詞 -> k o k o
名詞 -> k o t o

動詞五段わ行語幹 -> o m o
形容詞語幹-> o m o
```

図2：辞書用規則の例

このように辞書は

形態素カテゴリ -> 音韻列

という形で記述される。左辺の形態素カテゴリは、名詞・助詞などの非活用語の場合は品詞そのものにあたり、動詞・助動詞などの活用語の場合は語幹・活用語尾が別々のカテゴリとなる。

図2の規則は、トランスレータによって表1のLR表に変換される。

表 1：辞書の LR 表

state	h	o	k	m	N	t
0	sh1	sh2	sh3			
1		sh4				
2				sh5		
3		sh6, re1				
4					re0	
5		re4, re5				
6			sh7			sh8
7		re2				
8		re3				

図 2 に示したように、辞書用の規則は通常の文法とは異なり右辺には (辞書のレベルでの) 終端記号である音韻名しかこない。そのため、辞書用の LR 表は、動作表だけになっている。

また、辞書引きでは活用語の語幹を除いては未定義語の可能性などを考えると、形態素が完成したら、次の音韻に関係なく成功させなくてはならないという、特殊な性質を持つ。このため、通常の LR 構文解析法とは異なり、図 2 の 1 番上の規則を解析する場合、N を処理するところ (表 1 の状態 4 の N のエントリにあたる) では、先読みを待たずにレデュースを行うようにしている。レデュースを行うと、h o N という音素列で名詞というカテゴリが辞書引きできたことがわかる。

このような辞書引きの処理は、図 2 のトライ構造化辞書での辞書引きの処理と等価なものである。

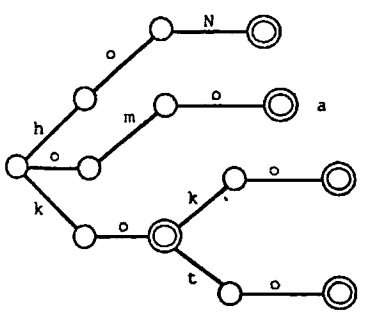

図 3：トライ構造化辞書の例

ここで、図の二重丸の部分は辞書引きが成功する場所を示している。トライ構造化辞書は、接頭部分の同じ音素列の部分は共有されるので (単語数が増えた場合に) 重複する処理を避けることができるため、テキスト入力の場合には効率の良い辞書の構成法として知られている [8] [2]。

本システムでは、LR 表で先読みされている音韻をその段階で予測される音韻であるとしてそれらのモデルに対してだけ並列に処理を進めることにより、音声入力に対応している [3]。

ところで、上記のようにして構成した辞書では、先に述べたように接頭部分の共通部分だけが共有される。このような性質は、通常の (誤りがないと仮定される) テキスト入力の場合には、入力が一意に定まり、途中で複数の経路を考える必要がないので、十分に効率のよい辞書引きを可能にする。しかし、音声入力のように複数の経路を考えなければならない場合には、接頭部分でない場合の共通部分 (図 3 の h o N の o と k o k o の o など) も同じ音声区間で処理する場合には共通に処理を行うことによってなるべく演算量を減らしたい。

そこで、本システムでは以下に示すアルゴリズムで、辞書引きを行う。

3.3 制御アルゴリズム

1. 初期化として辞書の根から予測されている音韻モデルに対してセルを作成し 1 桁目に設定する。各セルには初期スコアと対応する LR 表のエントリへのポインタを保持する。

2. (セルが設定されている) 音韻モデルの照合を行う。音韻モデルとの照合には、ビタビ・アルゴリズムを用いる。したがって、音韻モデルのある状態で経路の合流が起ると、そこまでの照合結果のよい経路のスコアをその状態でのスコアとして照合を進めていく。このとき、スコアを設定するのと同時に、その経路の起点となるフレームについても記憶しておく。

3. 音韻モデルの終了状態に達したら、その経路の起点となるフレームに設定されているセルを参照して、アクションがレデュースの場合は辞書引きを成功させる。シフトの場合は LR 表にしたがって、状態遷移を行う。そして、その状態で予測されているモデルに対してセルを作成し、次の桁に設定する。

4. 3 を (必要な) 全音素モデルに対して行ったら桁境界の処理を行う。桁境界では、その音韻に設定された最大のスコアを持つセルのスコアと次の桁の初期状態のスコアを比較して、大きい経路を残すようにする。

5. 2, 3, 4 の手順を全桁 (辞書中の最も長い単語の音韻数とする) に対して繰り返すという手順をフレームごとに行う。

ここで 4 の段階で同じモデルに対して複数のセルが (異なる経路、異なるスコアで) 設定されうる。図 1 の規則では 1 桁目の h と k に対して同じフレームで照合を終えた経路がそれぞれ存在すれば、そのフレームで 2 桁目の o には表 1 の状態 1 でのエントリと状態 3 でのエントリに対応する 2 つのセルが設定される。

3.1 で述べたように従来の One Pass DP アルゴリズムを用いる方法では、これらのセルから続けられる経路をすべて別々に扱っている。しかし、それでは、語彙数が多くなると、同じような音声区間の同じ音韻に対して非常に多くの候補を考慮しなくてはならなくなってしまう。したがって、本アルゴリズムではそういった場合には、スコアが最大のセルについてだけ、そのスコアを用いて次のレベルの処理を行う。そして 2 位以下のセルについては、1 位のセルと同じ経路をたどったものとしてスコアを更新していく。

このようなアルゴリズムを採用することによって、処理途中に保持しなければならない途中結果の空間的コストもそれらを別々に計算するための時間的コストも大幅に削減される。

しかし、このアルゴリズムでは、図4で、時刻t'においてスコアが$Ab > Cb' > Cb > Ab'$であるとすると、問題が生じる。

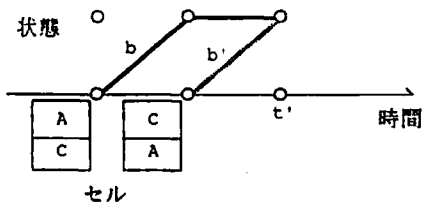

図4：最適性が失われる例

本来ならCを含む経路に関しては、Cb'の経路を採るべきだが、本アルゴリズムではCbで代用している。局所的にこのような処理を行うので、入力開始から終了まで常に1位であったような結果以外の辞書引き結果は必ずしも最適な経路を通って認識を行った結果とはならない。

このアルゴリズムでは、音韻の照合のための計算量は2でセルが設定される音韻モデルの数が単語数が非常に多くなってもたかだか全音韻モデル数であるので

単語の最大桁数×音韻モデル数×フレーム数

となる。セルの更新のための計算量は、

ビーム幅×フレーム数

となる。

必要な記憶容量は、音韻の照合のためには1つ前のフレームの値があればよいので、

単語の最大桁数×音韻モデル数

となり、セルの更新のためにはたかだか

ビーム幅×フレーム数

の分だけあればよい。これらの値は単語数には直接関係ないのでこのアルゴリズムは大語彙の辞書引きに向いた手法である。

One Pass DP アルゴリズムを用いてN-Bestな解を得る方法としては、[12]がある。この手法と比較すると、全てのパスを履歴ごとに管理する点は同じなので、セルに必要な記憶容量は等しく、N-Bestな解を得ることができるという点も同じである。しかし、音韻の照合に必要な計算量の点では、schwartsらの方法は、各経路を別々に計算するので

ビーム幅×フレーム数

だけ必要となる。一方、上で述べたように本手法では複数の経路を一つの経路で近似してしまっているので、その分最適性は失われていると考えられる。

また、さらに最適性を儀牲にして計算量を削減する方法も考えられる。それは桁を区別するのを止めてしまう方法で、図2の辞書を例にとるとomoの最後のoもkotoの最後のoも同じフレームから始まる経路の場合にはoの認識経路は同じとみなして処理を行う。

この手法では、音韻の照合のための計算量はたかだか、

音韻モデル数×フレーム数

であり、必要な記憶容量は

音韻モデル数

となる。

4 HMM

はじめに実験に用いた音声資料について述べる。音声資料は、単語音声と連続音声からなる。単語音声資料の話者は成人男性10名で、発声用テキストは音韻バランス単語集合WD-I(492語)[5]である。連続音声資料の話者は成人男性6名で上記の話者の一部である。連続音声の発声用テキストは疑問文などで11文ある。収録は簡易防音室で行ない、標本化周波数15 kHzでA/D変換を行なった。分析は、フレーム周期5 msごとに256点ハミング窓で切り出し、30次のFFTケプストラム係数を求め、これを14次のメルケプストラム係数に変換した。14次のメルケプストラム係数と、その時間方向の変化量、パワーの時間方向変化量の合計29個のパラメータを1つのコード帳にベクトル量子化した。メルケプストラム係数、時間方向の変化量、パワーの時間方向変化量の間の重みつけは、それぞれの分散が等しくなるようにした。コード帳のサイズは1024である。

HMMの音韻のモデル構造は、すべて4状態3ループとし、各状態から出る弧の組をタイドアークとした。単語認識実験に用いた音韻モデルの学習は、はじめに音韻ラベル付きの音声資料を用いて4回の繰り返し学習を行ない、つぎにこれを初期モデルとして、同じ音声資料に対してラベルなし学習(単語の発声記号にもとづいて連結した音韻モデルによる学習)を行なった(繰り返し回数4回)。話者openの実験とするため、ある話者をテストするための音韻モデルは、その話者以外の9名の音声資料を用いて学習した。連続音声認識実験用の音韻モデルは、単語認識実験用の音韻モデルを初期モデルとして、連続音声資料によって連結学習を行なったものである(繰り返し回数4回)。やはりテストする話者以外の5名の音声資料を用いて学習した。

5 単語認識実験

辞書引き部分に用いるアルゴリズムの有効性を検討するために、辞書引き部分だけを用いて、単語認識実験を行った。

5.1 実験の構成

4で述べた単語資料 WD-I (492 語) に対して、話者 open の単語認識実験を行った。ここでは、本稿で報告したアルゴリズムの最適性と効率のよさを検討するため、以下のような方式を用いて単語認識実験を行った。

1. 単語ごとにパターンを作成する方式
2. 本システムで採用した方式
   (a) 各音素モデル毎にビーム幅を 1 (1 位だけを考慮する手法とほぼ同じ) としたもの
   (b) そのフレームで最大のスコアから、スコアの差が 1×10^{-20} 以内のものを残したもの
   (c) (b) と同じ条件で桁を区別せずに探索を行ったもの

単語については、

単語 -> < a N e- >

のような規則で辞書を構成することによって、本システムの辞書引き部分を音声区間の開始フレームから一度だけ駆動することによって単語認識プログラムとして用いた。(< と > は、単語の前後の無音区間を示す。)

本実験では、単語を構成するために表に示した音韻を用いた。この音韻 1 つに対して、HMM が 1 つ用意されている。

表 2 : 音韻モデルの種類

a	i	u	e	o	N	b	d	g	m
n	r	s	sh	h	f	y	w	p	t
k	ch	ts	j	z	by	dy	gy	my	ny
ry	py	hy	ky	q	a-	i-	u-	e-	o-
<	>								

音声資料の終了フレームで辞書引きが終了した経路のうちスコアが最も高い経路を認識結果の単語とした。

5.2 実験結果

各方式の単語認識率を表 3 に示す。

表 3 : 話者 open の実験結果 (単語認識率) (%)

方式	平均	最低/最高
1	96.7	94.5 98.2
2 (a)	61.2	54.3 68.5
2 (b)	94.9	91.5 97.8
2 (c)	88.0	82.7 92.5

これらは全て同じ音韻モデルを用いているので、ビタビ・アルゴリズムを用いた場合の最良の結果が、方式 1 の認識率になる。十分に枝刈りのための敷居値を大きくした 2(b) の方式では、方式 1 に近い認識率が得られており、最適性は余り失われていない。2(c) の方式では、敷居値は 2(b) と同じであるが、認識率はさらに劣化している。

次に各方式を用いて、同じ 1 名の話者の "a N e-"(暗影) という単語を認識するのに必要な認識時間と、認識するのに必要となったセル数を表 4 に示す。表の認識時間は SONY NEWS-3860 を用いて得た時間である。使用セル数は、入力の開始フレームから終了フレームまで認識を進めるうえで必要となったセル総数を数えあげた。

表 4 : 話者 open の実験結果 (認識時間など)

方式	認識時間 (秒)	使用セル数
1	24.9	—
2 (a)	4.8	85246
2 (b)	6.6	204636
2 (c)	0.9	59068

1 の方式に比べると、どの方式も認識時間が短くなっており効率がよくなっていることが示されている。2(a) の方式と 2(b) の方式を比較すると、使用セル数は 2(b) が 2(a) の 2.4 倍となっており、ビーム幅がその分広くなっていると考えられるが、認識時間は 1.3 倍とビーム幅が広がったほどには増えていない。これは、3.3 で述べたようにビーム幅が広がってもセルの更新のための計算時間は増えるが音韻照合のための計算時間はほとんど増えないことを示している。2(c) は 2(b) に比べて使用セル数は、1/3.5 であるが、3.3 で示したように音韻照合のための計算時間も少なくてすむため、認識時間は 1/7.3 となっている。

6 拡張 LR 構文解析法を用いた辞書引き結果の接続

6.1 拡張 LR 構文解析法

LR 構文解析法を用いて自然言語を構文解析する方法としては、拡張 LR 構文解析法 (富田法) [13] がよく知られている。拡張 LR 構文解析法は LR 表での動作の曖昧さ (コンフリクト) や多品詞語によって生じる曖昧さに対して、複数の動作を並列に進めることによって複数の解析を行う手法である。入力を左から右に横型探索で探索を進めるので、フレーム同期の処理に適している。

図 5 に拡張 LR 構文解析法で解析することのできる文法規則の例を示す。

結果 -> 無音1 文 無音2
文 -> 文節
文 -> 文節 文
文 -> 文節 ポーズ 文
文節 -> 名詞
文節 -> 名詞 語尾1
文節 -> 修飾語 語尾1
文節 -> 修飾語
文節 -> 動詞 語尾2

図5：文法規則の例

この規則はトランスレータによって表5のLR表に変換される。

しかし、3の手法によって得られる辞書引き結果をフレームごとに同期をとって処理する場合には、通常のテキスト入力の場合には考慮されない、以下に上げる現象が起こる。

- 同じ単語に対して開始フレームと終了フレームが少しずつ異なる認識結果が得られる。
- 同じ音声区間で、同じ(文法での)終端記号に対して様々な認識結果が得られる。

本システムでは、基本的には拡張LR構文解析法に基づいて曖昧さに対して複数の候補を考慮し、その数が一定以上になった場合には枝刈りを行うことで対処する。したがって、曖昧な候補が増加しても解析途中の候補の数は一定に抑えられる。しかし、上記の2点、及びその組合せによって生じる認識結果を別々に扱うと、結局は同じ解析結果になる候補を重複して考慮することになる。その結果、認識精度が悪化し、重複する構文解析処理を行うため処理時間も無駄になる。したがって、なるべく処理時間を抑えて、認識精度をよくしようとするならこれらの問題を解決しなければならない。

本システムでは、これらの問題にできるだけ対応するために以下で述べる機構を導入した。

6.2 同一結果となる候補の回避

解析中に同じ候補を重複して考慮することは、辞書引きを3.3で述べたアルゴリズムで行っているため避けることができる。例えば、hoNgaという入力のとき、hoNという形態素の辞書引きが異なるフレームで終了して複数の経路となっても、gの音韻照合のところで合流すれば、もっともよいスコアの経路1つにまとめられる。

6.3 構文解析の重複の回避

音声認識システムでは、音韻の認識にかかる時間に比べれば構文解析に要する計算時間はわずかであるが、意味解析などの処理をさらに行うことなどを考えると、構文レベルでもなるべく無駄な処理は避けたい。

たとえば、同一フレームで構文的なカテゴリが等しいいくつかの単語の辞書引きに成功したような場合には、それらの辞書引き結果は、音韻系列のスコアなどは異なるが、構文のLR表を参照し状態を遷移して次の形態素カテゴリを予測するという処理は全く同じである。したがって、これらの処理を何度も行うのは明らかに無駄である。

しかし、これらを構文レベルで完全に同一視することはできない。なぜなら、音韻系列のスコアや、音韻連鎖や文法カテゴリの連鎖に関する確率モデルや、辞書引き結果から得られる意味的素性など、処理途中にそこまでの(特に構文以外のレベルの)解析結果を用いて枝刈りをする場合には、結局個別に処理する必要があるからである。

そこで、辞書引きアルゴリズムで用いるセル(以下、区別のため音韻セルと呼ぶ)とは別に、LR構文解析に関して同じ解析履歴となる候補の情報を保持するための構文セルを用意する。構文セルには、対応する(構文レベルの)LR表のエントリへのポインタと、そのときのスタック・トップへのポインタを保持させる。

LR構文解析法ではLR表とスタックとの対応で解析を進める。スタックについても、スタックの操作の重複やスタックの状態の保持のための記憶容量を考慮すると、スタックが全く同じになる候補に関しても、スタックに関する情報を共有したほうがよい。そこで、スタックに関してはスタック木と呼ぶデータ構造を導入する。

以下、表5に示したLR表を用いて解析を進めていくときに、どのように構文セルとスタック木が構成・利用されるかを示す。

1. まず、LR表の状態0から解析を始める。スタックには状態0が積まれる。状態0では、無音1が予測されているので、無音1に対する構文セルが作成され、辞書引き処理が開始される。(図6)

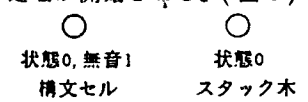

図6：構文セルとスタック木を用いた構文解析過程(1)

2. 解析が進んで無音1の辞書引きが終了したら、その結果が構文セルに受渡され、LR表にしたがって状態1に遷移し、スタックには構文カテゴリ無音1と新しい状態1をプッシュする。

状態1では名詞・修飾語・動詞が予測されているので、それに対応して構文セルが作成される。そして、新しい辞書引き処理が開始される。(図7)

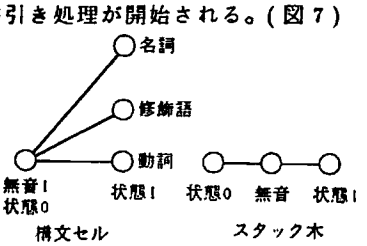

図7：構文セルとスタック木を用いた構文解析過程(2)

表 5: 構文レベルの LR 表の例 (一部)

	動作表									行先表		
state	無音 1	無音 2	ポーズ	名詞	助詞	語尾 1	語尾 2	修飾語	動詞	$	結果	文	文節
0	sh1										2		
1				sh3				sh4	sh5			6	7
2										acc			
3		re6		re6	sh8	sh9		re6	re6				
4		re9		re9		sh10		re9	re9				
⋮													

このようにして構文セルは木構造状に作成される。ここで、2 の処理が行われた次のフレームでも無音 1 の辞書引きが終了する可能性があるが、それらの辞書引き結果は、スタックや LR 表を参照することなく、次の名詞・修飾語・動詞のセルに移り、新しい辞書引き処理を開始させる。

また、処理が進んでレデュースがおこると、スタックをポップさせることになるが、スタック木では、ポップは根の方へ枝をたどる処理となる。

このような処理方法をとることによって、同じ構文処理は一度しか行わずにすむ。また、音韻セルはその辞書引き処理を開始させた構文セルへのポインタを保持することによって、その辞書引き結果を構文解析部に受渡せるようにしている。

7 文認識実験

本稿で述べた連続音声認識システムの動作確認のために、4 で述べた連続音声資料を用いて、話者 open の文認識実験を行った。文法は、図に示したものを用い、形態素は 11 文に現れる 37 個だけから辞書を作成した。

辞書引きには、桁を区別する方式を用いた。枝刈りは、各フレームで最大のスコアから、スコアの差が 1×10^{-20} 以内のものを残すように行った。

音声資料の終了フレームで、アクセプトする経路のうちスコアが最も高い経路を認識結果の文とした。

実験は、2 人の話者に対して 11 文ずつ行った。その結果 1 位の候補が正解となったのが、16 文で、2 位が正解となったのが 2 文、7 位が正解となったのが 1 文、正解が途中で枝刈りされてしまったのが、3 文であった。図に認識誤りの例を示す。

8 おわりに

連続音声認識システムにおいて、自然言語処理と音声認識処理の統合化の基礎となる手法について報告した。ここで提案した手法は、音声のパラメータ系列から直接辞書引きを行い、さらに構文的な制約も加えながら、最適性の高い音韻系列を複数解得ることができる。

辞書引きアルゴリズムは、従来の手法から少し最適性を犠牲にすることで、大規模な辞書も、計算量をそれほど増やすことなく扱うことを可能にしている。

今後は日本語の大規模な文法を開発し、大規模な文認識実験を行いながら、文脈依存 HMM や音韻や形態素カテゴリの連鎖に関する統計情報などを組込むことによって、システムの性能向上を目指す予定である。

また、辞書の音韻パープレキシティと本稿で報告した辞書引きアルゴリズムの効率・認識率の関係や、文のレベルでの音韻パープレキシティと本稿で報告した連続音声認識手法の関係についての検討も今後の課題であろう。

参考文献

[1] 岡田美智男, アクティブチャート解析法に基づく One-Pass アルゴリズムの構文制御について. 信学会技術報告, SP90-24:39-46, 1990.

[2] 上脇正, 田中穂積. 辞書の trie 構造化と熟語処理. , *Logic Programming Conference'85*, 329-340 ページ, ICOT, 1985.

[3] 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, 31(3):472 - 480, 3 1990.

[4] 速水悟, 田中和世, 太田耕三. 音素片ネットワークによる音声の音響的変動の記述と単語認識実験. 信学論 *(D)*, J71-D(2):265-273, 2 1988.

[5] 速水悟, 田中和世, 横山晶一, 太田耕三. 研究用音声データベースのための VCV/CVC バランス単語セットの作成. 電総研彙報, 49(10):803-834, 1985.

[6] 堀内靖雄, 伊藤克亘, 田中穂積. 拡張 LR 構文解析アルゴリズムによる未定義語を含む日本語文の構文解析. , 情報処理学会 第 *40* 回全国大会, 325 - 326 ページ, 1990.

[7] 南泰浩, 中川正雄. Trigram モデルを用いた複数候補を求めるフレーム同期型 HMM 連続音声認識. 信学論 *(D-II)*, J73-D-II(9):1383-1391, 9 1990.

[8] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *Data Structures and Algorithms*. Addison-Wesley, 1983. 邦訳: 大野義夫訳, データ構造とアルゴリズム, 培風館, 1987.

正解：/ o m o sh i r o i t o o m o i m a s u k a /
(面白いと思いますか)
結果：/ o- k i- o m o sh i r o i t o o m o i m a s u k a /
(大きい面白いと思いますか)

正解：/ s o r e n i w a d o N n a k o t o g a k a i t e a r i m a s u k a /
(それにはどんなことが書いてありますか)
結果：/ s o r e n i w a d o N n a s o n o t o g a k a i t e a r i m a s u k a /
(それにはどんなそのとが書いてありますか)

正解：/ a n a t a w a s o n o h o N o y o m i m a sh i t a k a /
(あなたはその本を読みましたか)
結果：/ a n a t a s o n o o y o m i m a sh i t a k a /
(あなたそのを読みましたか)

図 8: 認識誤りの例

[9] J. S. Bridle, M. D. Brown, and R. M. Chamberlain. An algorithm for connected word recognition. In *Proc. ICASSP-82*, pages 899-902, IEEE, 1982.

[10] K. F. Lee. *Automatic Speech Recognition : The Development of the SPHINX System.* Kluwer Academic Publishers, 1989.

[11] H. Ney, D. Mergel, A. Noll, and Paeseler. Data-Driven organization of a Dynamic Programming beam search for continuous speech recognition. In *Proc. ICASSP-87*, pages 833-836, IEEE, 1987.

[12] R. Schwarts and Y. L. Chow. The N-best Algorithm: An efficient and exact procedure for finding the N most likely sentence hypotheses. In *Proc. ICASSP-90*, pages 81-84, IEEE, 1990.

[13] M. Tomita. An efficient augmented-context-free parsing algorithm. *Computational Linguistics*, 13(1-2):31-46, 1987.