TR-I-0325

文脈自由文法を用いた連続音声認識
Continuous Speech Recognition Based on Context-Free Grammars

北 研二　　森元 逞
Kenji KITA　　Tsuyoshi MORIMOTO

1993.3

概要

本レポートでは、文脈自由文法を用いた連続音声認識に関する成果について報告する。予測 LR パーザと HMM 音韻認識を統合化した HMM-LR による文節音声認識，2段階 LR 法を用いた文音声認識，統計的言語情報を用いた HMM-LR による文連続音声認識などについて説明している。

目 次

1 文脈自由文法

自然言語をコンピュータで扱うためには、言語現象を形式的に記述する体系が必要である。文脈自由文法は、言語の統語 (syntax) 的な構造を記述するために、自然言語処理の分野でよく用いられているモデルである [1, 2]。この理由として、多くの言語現象が文脈自由文法によって表現されること、文脈自由文法に対しては効率的な解析アルゴリズムが開発されていることなどがあげられる。

形式的には、文脈自由文法は 4 つ組 $< V_N, V_T, P, S >$ によって定義される。ここで、V_N は非終端記号 (nonterminal) の集合、V_T は終端記号 (terminal) の集合、P は生成規則 (production) の集合、S は開始記号 (start symbol) である。非終端記号とは「名詞句」とか「動詞句」などの抽象的な文法カテゴリを表す記号であり、終端記号は個々の単語 (音韻とか音節であってもかまわない) を表す。生成規則は文法カテゴリあるいは単語間の階層的な関係を記述するものであり、$A \rightarrow \alpha$ (A は非終端記号, α は非終端記号あるいは終端記号から成る記号列) の形をした規則である。生成規則 $A \rightarrow \alpha$ は、左辺にある A という記号を右辺にある α という記号列に書き換えることを意味しており、書き換え規則と呼ばれることもある。また、開始記号は特殊な非終端記号であり、「文」という最も上位の文法カテゴリを表す記号である。

生成規則を何回か適用することにより記号列 α が β に書き換えられるとき、$\alpha \Longrightarrow \beta$ と書き、α は β を導出する (あるいは β は α から導出される) という。文脈自由文法によって定義される言語は、開始記号 S から導出される終端記号列の集合である。また、生成規則適用の際に最も右 (左) 側の非終端記号を書き換えるような導出は、特に最右 (左) 導出 (rightmost or leftmost derivation) と呼ばれる。文法によっては、ある終端記号列に対していく通りもの最右導出が存在することがある。このような文法は曖昧 (ambiguous) であるという。

図 1 は、文脈自由文法の例を示している。図 1 の文法では、
$V_N = \{S, NP, N, P, V\}$、$V_T = \{e, o, u, k, r\}$ であり、開始記号は S である。また、各生成規則には (1) から (8) までの番号が付けられている。この文法は、音韻単位の音声認識で用いられることを想定しているために、終端記号は単語ではなく音韻となっている。

(1)	S	$\rightarrow$	$NP\ V$
(2)	S	$\rightarrow$	V
(3)	NP	$\rightarrow$	N
(4)	NP	$\rightarrow$	$N\ P$
(5)	N	$\rightarrow$	$kore$
(6)	P	$\rightarrow$	o
(7)	V	$\rightarrow$	$kure$
(8)	V	$\rightarrow$	$okure$

図 1: 文脈自由文法の例

2 LR パーザ

記号列が与えられたとき、その記号列が文法からどのようにして導出されるのかを調べることを構文解析と呼んでいる。また、構文解析を実行する処理系はパーザ (parser) と呼ばれる。LR パーザ [2] は、もともとプログラミング言語処理系の分野で開発されたアルゴリズムであり、曖昧な文法を扱うことができない。しかし、後で述べるように、LR アルゴリズムを一般の文脈自由文法にまで拡張したアルゴリズム (拡張 LR アルゴリズム) が最近開発されており [3, 4]、自然言語処理の分野でもよく使われるようになった。拡張 LR アルゴリズムは、自然言語の文法を処理する上で最も効率的なアルゴリズムであるといわれている。

LR パーザは LR 解析表と呼ばれる表を参照しながら解析を進める。LR 解析表は文脈自由文法から解析に必要な情報をあらかじめ計算して作られた表で、文法から自動的に生成することができる (LR 解析表の作成については (Aho,1986) [2] などを参照)。図 1 の文法からは図 2 に示すような LR 解析表が作られる。図の左側の部分は動作表と呼ばれ、右側の部分は行先表と呼ばれている。このうち、動作表はパーザの現在の状態 s と入力記号 a からパーザが次に取るべき動作 $ACTION[s, a]$ を決定するのに用いられる。パーザの動作には、移動 (shift)、還元 (reduce)、受理 (accept)、誤り (error) の 4 種類がある。図 2 では、s で始まるものが移動動作であり、r で始まるものが還元動作である。また、acc で示されているものが受理を示し、図中空欄であるものは誤りであることを示している。動作表の記号欄の\$は入力の終りを示しており、入力記号列の最後には\$が付いていると仮定している。

状態	動作表						行先表				
	e	*o*	*u*	*k*	*r*	$	*S*	*NP*	*V*	*P*	*N*
0		*s*3		*s*4			1	5	2		6
1						*acc*					
2						*r*2					
3				*s*7							
4		*s*8	*s*9								
5		*s*3		*s*11					10		
6		*s*13, *r*3		*r*3						12	
7			*s*14								
8					*s*15						
9					*s*16						
10						*r*1					
11			*s*9								
12		*r*4		*r*4							
13		*r*6		*r*6							
14					*s*17						
15	*s*18										
16	*s*19										
17	*s*20										
18		*r*5		*r*5							
19						*r*7					
20						*r*8					

図 2: LR 解析表の例

LR パーザは、入力された記号列がどこまで処理されたかという入力ポインタと状態記号に対するスタックを持っている。パーザは以下のようにして解析を行なう。

1. 入力ポインタを入力記号列の先頭にセットし、スタックに初期状態記号である 0 をプッシュする。

2. スタック最上段の状態 s と入力ポインタの指す記号 a に対して、$ACTION[s, a]$ を調べる。

- $ACTION[s, a]$ が "移動" であれば、入力ポインタを次の記号に移動することを意味している。移動動作では、s の後に示されている番号をスタックにプッシュし、入力ポインタを一つ進め次の入力記号を指すようにする。
- $ACTION[s, a]$ が "還元" であれば、スタックの上部にある記号を生成規則によりまとめあげていく動作であることを意味している。還元動作では、rの後に示されている番号nで示される生成規則の右辺にある文法記号の数だけスタックから状態記号をポップする。次に、スタック最上段の状態 s' と n 番目の生成規則の左辺にある文法記号 A とから行先表を調べて、$GOTO[s', A]$ をスタックにプッシュする。なお、還元動作では入力ポインタは進めない。
- $ACTION[s, a]$ が "受理" であれば、入力された記号列が文法により無事解析されたことを意味している。
- $ACTION[s, a]$ が "誤り" であれば、入力された記号列は文法から受理されない (導出されない) ことを意味している。

3. 再び 2 に戻り、解析を続行する。

文法が曖昧であるときには、LR 解析表には多重に定義された動作項 (multiple entry, conflict) が出現する。図2では状態 6 、入力記号 o の箇所で $s13$ と $r3$ が多重に定義されている。通常の LR アルゴリズムは曖昧な文法を扱うことができないが、動作表に複数の動作が定義されている場合には各動作を同時並行的に行なうことにより、文法の曖昧性にも対処することができる。また、このように拡張された LR アルゴリズム (generalized LR parsing) ではスタックをグラフ構造化し (graph-structured stack)、スタックの分岐および併合を行なうことにより解析処理の効率化を行なうことができる [3, 4]。

図3には、図1,2で示された文法および LR 解析表を用いて、入力文 "$koreokure$" を解析する様子が示されている。入力記号 e を処理し $r5$ という動作を行なったあと、スタック最上段の状態が 6 となり、ここで $s13$ と $r3$ という多重定義欄が現われる。各動作を並行して行なうために、スタックの分割が行なわれている。この図では、わかりやすいように、スタックは多重化して表されており、グラフ構造化は行なわれていない。図中、四角の箱で書かれているものがスタックの中身であり、括

弧内に状態番号が示されている。状態番号の左側には、対応する文法の記号が示されている。

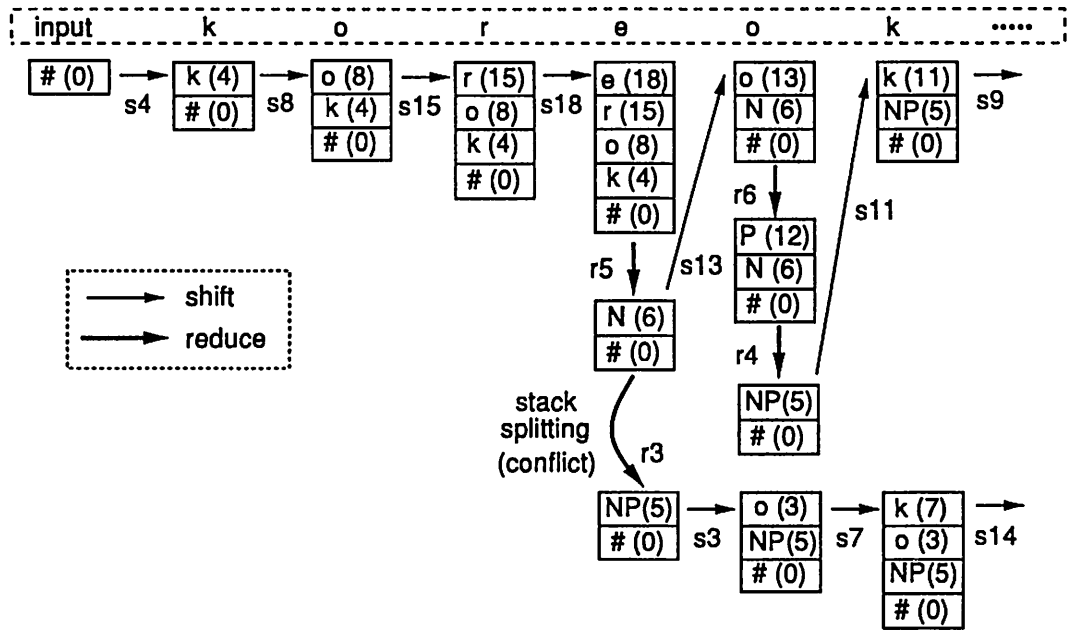


図 3: 曖昧な文法に対する LR パーザの解析の様子

3 HMM-LR 音声認識

3.1 方式概要

HMM-LR 音声認識 [5, 6] は、音韻ベースの隠れマルコフモデル (HMM) と LR パーザを統合化した音声認識方式である。HMM-LR 音声認識では、文法から得られる予測的な情報を用いて音声認識の探索空間を縮小するだけでなく、LR パーザの中から HMM 音韻モデルを直接駆動して認識を行なうために、音声認識と言語処理の間に音韻ラティス等の中間的なデータを介する必要がなく、高精度でかつ効率的な大語彙連続音声認識システムを構築することができる。

通常の文字列を対象とする自然言語処理では、解析すべき記号列 (入力文字列) があらかじめ与えられている。このために LR パーザでは次の入力記号を参照して、パーザが次にとるべき動作を決定することができる。しかし、音声認識では逆に入力されたものが何かを求めるのが問題であり、認識の途中では次の入力記号が何かというのは全くわからない。このため HMM-LR 音声認識では、LR パーザはそれまでに認識されてきた音韻系列から次にくる音韻を予測しながら認識動作を進める。このように LR パーザは予測的に用いられるため、特に予測 LR パーザと呼ばれている。

予測 LR パーザによる音韻予測は LR 解析表の動作表を用いて行なわれる。動作表は状態と入力記号 (音韻) によって表されている 2 次元の表であるので、パーザの状態 (スタック最上段の状態) が与えられればパーザがその状態でどのような音韻を予期しているのかがわかる。より正確にいえば、状態に対する動作欄の中で移動動作 (shift) の指定されている音韻が次にくると予測される。即ち、状態 s から予測される音韻の集合は次により求められる。

$$Prediction(s) = \{x | Action[s, x] = \text{“}shift\text{”}\} \tag{1}$$

図 4 には、図 1,2で示された文法および LR 解析表を用いたときの予測 LR パーザによる音韻予測の様子が示されている。状態 0 で移動動作の指定されている音韻は o と kなので、状態 0 からは二つの音韻 o と kが予測される。それぞれの予測に対しスタックの分割が行なわれている。k の予測の次には状態は 4 となり、音韻 o と u が予測される。この場合もやはりそれぞれの予測に対しスタックの分割が行なわれる。

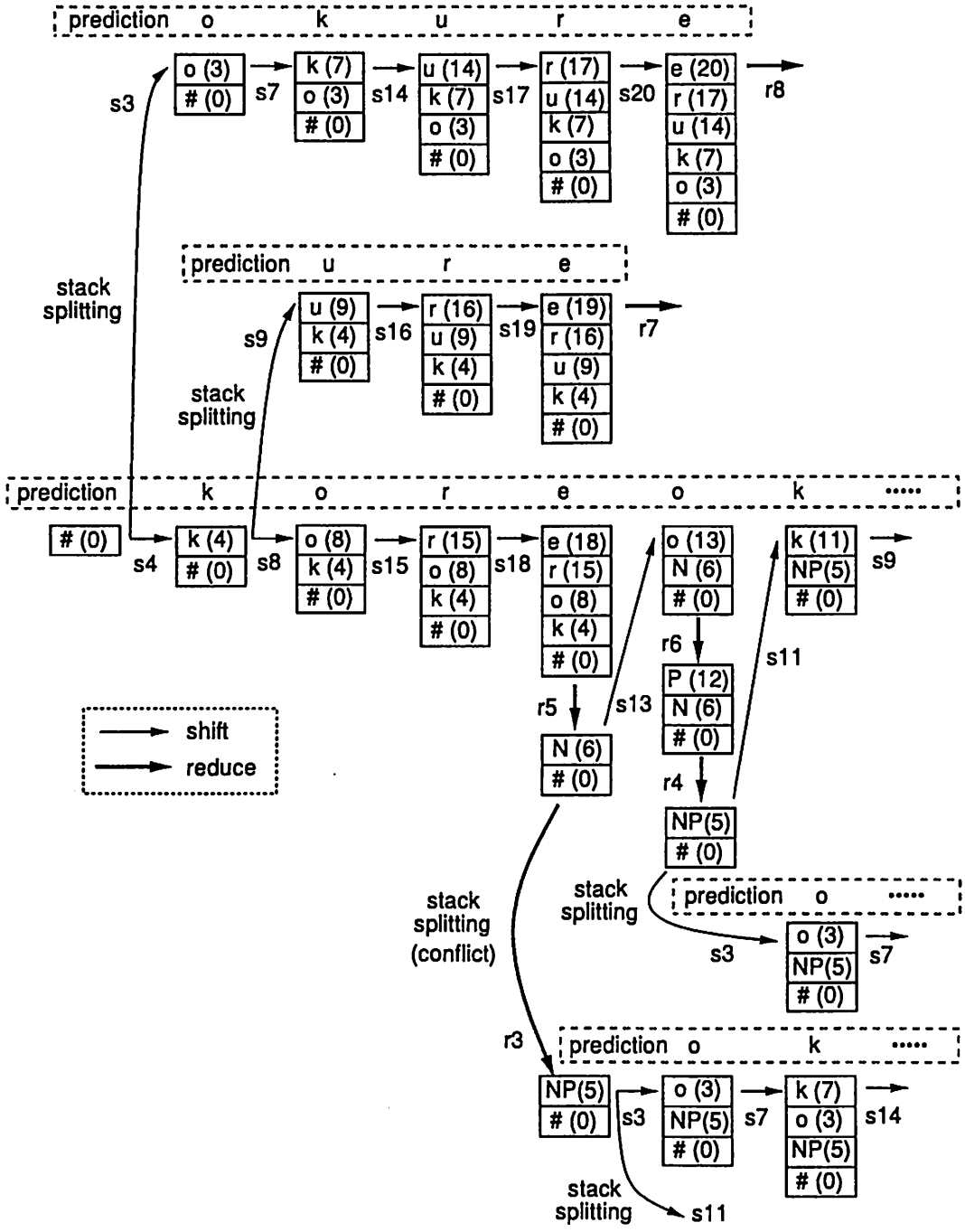


図 4: 予測 LR パーザによる音韻予測

予測LRパーザによる音韻予測と隠れマルコフモデル(HMM)による音韻照合を交に繰り返すことにより連続音声認識が実現できる(図5参照)。これが、HMM-LR声認識の基本的な原理である。より詳しい認識動作は以下のようになる。

1. 予測LRパーザにより、その時点までに既に認識された音韻系列から文法上で次に接続しうる(文頭であれば文頭にくることが可能な)音韻を予測する。音韻予測の際に、LR解析表の現在の状態欄に受理動作が指定されていれば、その音韻系列は認識候補として残される。

2. 各音韻に対して継続時間長の最小値および最大値等の統計情報があらかじめ求められており、これらの値を使って照合する音声区間を決定する。照合する音区間は現在までに認識された音韻系列と予測された音韻の最小継続時間長の和を始端、最大継続時間長の和を終端とする区間である。

3. HMM音韻照合部を駆動し、既に認識された音韻系列の音韻モデルに予測された音韻のモデルを連結して音韻照合を行ない、照合スコアを求める。ここで照合スコアとは、照合音声区間が音韻モデルから生成される尤度であり、forwardアルゴリズムあるいはViterbiアルゴリズムによって計算される。

4. 照合に成功したすべての音韻に対して、並行して音韻連鎖の枝を伸ばしていく。実際には音韻連鎖の枝を伸ばす過程において解析する候補の数が増加してくるので、照合スコアがある一定値以下の場合は枝刈りするというビームサーチを行ない、解析する候補数を削減する。

5. 再1に戻り、認識を続行する。

最終的に、照合がすべて終った段階で照合スコアの高い第n候補までを認識結果して出力する。このようにHMM-LR音声認識では音韻ラティス等の中間形式をさずに認識が進むので、高い認識性能が得られる。

2 文節音声認識

ここでは、上に述べたHMM-LR音声認識を日本語の特定話者の文節認識に適用た場合の性能評価について述べる。

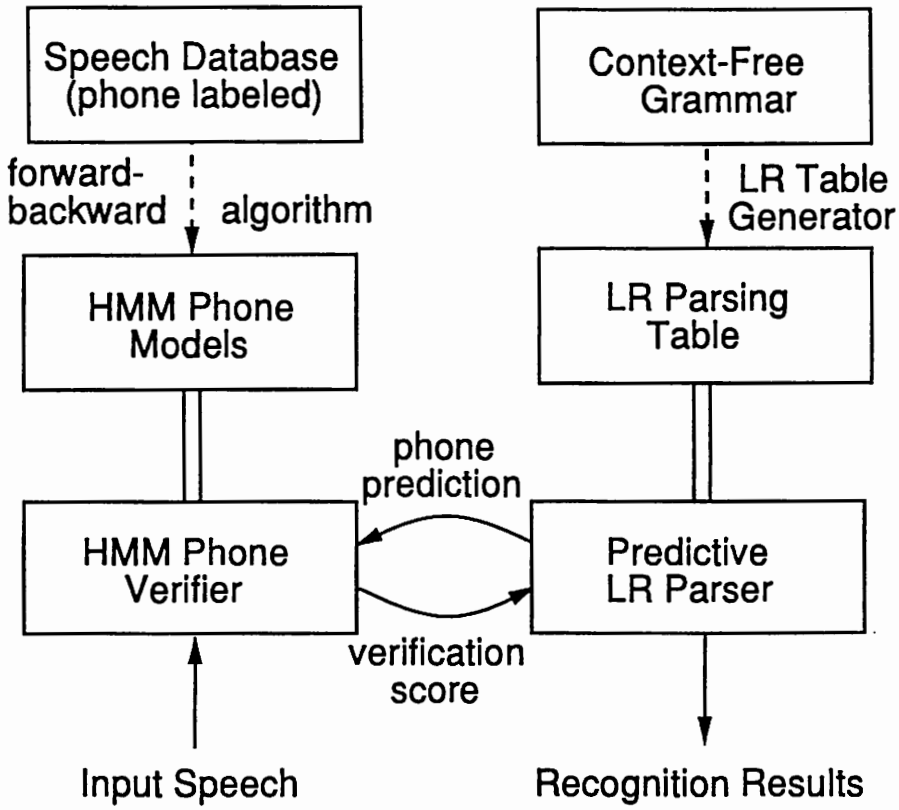


図 5: HMM-LR 音声認識の模式図

3.2.1 タスクおよび文法

評価に用いたタスクは「国際会議に関する問い合せ」を想定して作られたモデル会話A，B，1〜5であり、この中には全部で３５３個の文節が含まれている。また、評価実験で用いた文法は、１９７３個の生成規則から成る。生成規則は、文節内部の構文規則あるいは語彙規則に分類することができる。音声認識の単位が音韻であるため、語彙規則には各単語の音韻表記が書かれている。なお、この文法の語彙数は７４４である。

音声認識の性能は、認識時に使われる文法によって大きく左右される。非常に制限された文法を用いれば高い認識率が得られるし、多くの言語現象を扱うことのできる複雑な文法を用いれば認識率は低下する。また、単語数が増えればそれだけ混同する単語数も増えるので、やはり認識率は低下する。音声認識で用いられている文法がどれだけ複雑かということを評価する尺度として、パープレキシティ(perplexity)と呼ばれるものがよく用いられている。これは情報理論的な意味での平均分岐数であり、この値が大きいものほど文法として複雑であることになる。評価実験に用いられた文法の１音韻あたりのパープレキシティは３．５７である。

3.2.2 HMM 音韻モデル

認識実験で用いたHMMの音韻モデルは、音声の特徴量をベクトル量子化して扱う離散型のモデルである。音韻モデルの学習には、約5,000単語の単語発声の音声データを用いている。音声の特徴量としては、スペクトル、差分ケプストラム、パワーの３種類を用いており、各々別々のコードブックで表現するセパレートベクトル量子化の手法を使っている。また、HMMの各状態の継続時間長の分布を正規分布で表現した継続時間長制御を行なっており、これにより高精度の音韻認識を実現している。

3.2.3 認識結果

ビーム幅が１００および２５０の場合について、評価実験を行なった。結果を表1に示す。いずれの場合も１位に認識された文節は９２%以上、５位までの累積認識率は９９%以上であり、HMM-LR 音声認識の文節認識での有効性を示しているといえる。

表 1: 文節認識率

認識順位	累積認識率	
	ビーム幅 = 100	ビーム幅 = 250
1	92.4	92.6
～2	98.6	98.9
～3	99.4	99.7
～4	99.4	99.7
～5	99.4	99.7

3.3 大語彙音声認識

3.3.1 実験概要

文節音声認識で用いられた文法に語彙を追加することにより、大語彙化した場合の認識率について調査した [7]。このため、ＡＴＲ対話データベース中の頻出語と新明解国語辞典の重要語（固有名詞，数詞，機能語，間投詞は除く）に対して単語の音韻表記を作成した。表 2にそれぞれの単語数、異なり単語数を示す。

表 2: 単語数、異なり単語数

	単語数	異なり単語数
文節文法	744	497
ＡＴＲ対話データベース頻出語	2,050	1,747
新明解国語辞典重要語	5,140	4,445
計	7,934	6,689

認識実験では、ＡＴＲ対話データベース頻出語および新明解国語辞典重要語（計７１９０語）から任意に語彙を抽出し、それを文節文法に付け加えたものを用いた。このようにして、１０００語ずつ語彙を増やし（最大７９３４語まで）、文節認識実験を行なった。ただし、音声認識率は文法に含まれている語彙によって変わるので、各語数に対して、ランダムな語彙抽出と認識実験を 3 回行ない、その平均値をその語数に対する認識率とした。

また、音声認識では言語の統計的な情報を用いることにより、認識精度の向上が期待できるので、日本語の音節の3つ組確率 (trigram) を併せて用いた場合についても実験を行なった。

HMM-LR 音声認識システムでは、言語の構文的な情報は LR 構文解析表の中に埋め込まれている。大語彙化した場合、LR 構文解析表の大きさがどうなるかも、システムにとって重要な要因である。単語数の増加にともなう LR 構文解析表の大きさについても調査した。

3.3.2 実験結果

図6は、単語数と文節認識率の関係を示している。単語数の増加にともない、認識率も低下しているが、第5位までの累積認識率ではいずれも 94〜95 %以上の認識率を達成している。正しい文節が5位程度までに認識されていれば、あとの文単位での言語処理の段階で正しく認識されることが期待できるので、HMM-LR 音声認識は数千語規模の大語彙化に対しても十分な認識性能を持っていると考えられる。

図7は、単語数と LR 構文解析表の大きさ（状態数）の関係を示したものである。単語数の増加にともなう LR 構文解析表の状態数の増え方はほぼ線形である。これは、HMM-LR 音声認識システムが単語数の増加に対して耐性があることを示している。

図8は、単語数と文法の音韻あたりのパープレキシティの関係を示したものである。音韻パープレキシティは単語数が増えても、ゆるやかに増えるだけである。これは、HMM-LR 音声認識システムが、8000 語以上に単語数を増やしても十分な認識性能が得られる可能性を示しているといえる。

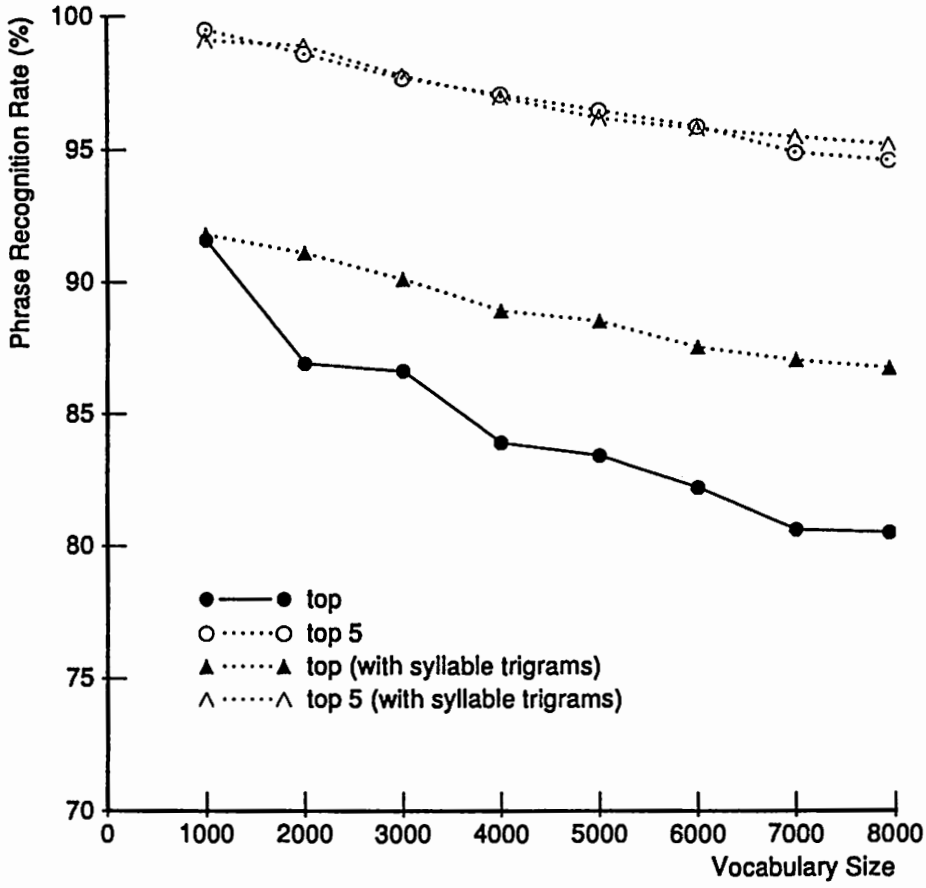


図 6: 単語数と文節認識率の関係

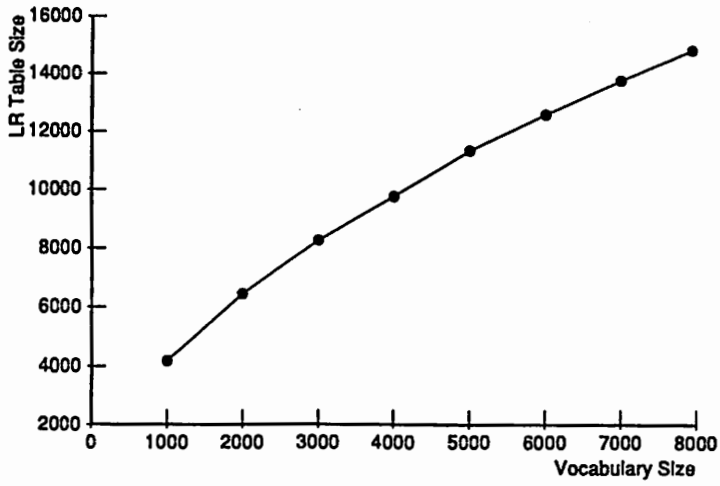


図 7: 単語数と LR 解析表の大きさの関係

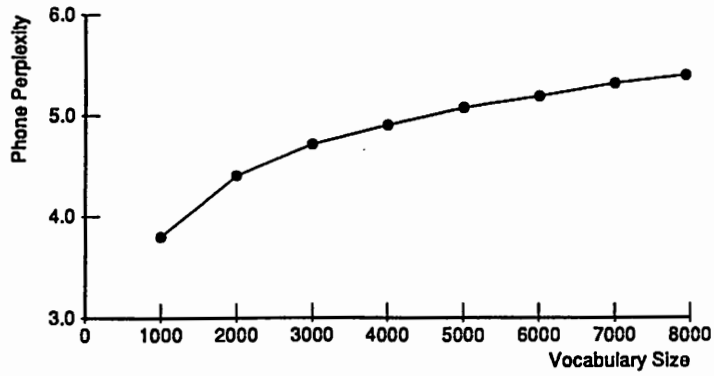


図 8: 単語数と音韻パープレキシティの関係

4　2段階LR法を用いた音声認識

前項では、HMM-LR音声認識を日本語の文節認識に適用した場合の結果について記した。文節認識の場合には、文節内の制約を記述した文節内文法を用いていたが、ここでは文節内文法に加え、更に文節間の制約を記述した文節間文法を用いた日本語の文の認識について説明する。ここで説明する方法は、文節間の制約と文節内部の制約を段階的に用いるものであり、2段階LR法[8]と呼ばれる方法を用いている。

4.1　方式概要

4.1.1　文節間文法および文節内文法

2段階LR法による文認識の基本的な動作はHMM-LRと同様であり、文節間予測LRパーザと文節内予測LRパーザの2つを用いて、文節カテゴリ(名詞句, 動詞句等)の予測と音韻の予測を段階的に行ないながら認識処理を進める。2段階LR法では、ある文節カテゴリが予測されたときに、その文節カテゴリに属するような音韻系列だけを文節内LRパーザが生成する必要がある。このために、文節内LR解析表に拡張を施している。

通常のLR解析表は、動作表(action table)と行先表(goto table)の2種類の表から構成されているが、文節内LR解析表にはもう1つの表（以下では、文節行先表(goto-phrase table)と呼ぶ）が新たに付け加えられている。文節行先表は、文節カテゴリとLRパーザの状態の組から成るリスト構造であり、ある特定の文節カテゴリを解析するためには、どの状態をパーザの初期状態とすればよいかが記されている。

図9と図10に文節間文法と文節内文法の例を示す。この例では、文節間文法の終端記号は名詞句(NP)と動詞句(VP)であり、「文は名詞句と動詞句から成るか、あるいは動詞句だけから成る」ということを記述している。また、文節内文法では「文節は名詞句かあるいは動詞句である」ということを記述しており、名詞句と動詞句の内部の構成規則を記述している。また、図11と図12に文節間LR解析表と文節内LR解析表の例を示す。図12中の文節行先表では、名詞句(NP)に対しては初期状態を1に、また動詞句(VP)に対しては初期状態を2にすればよいことが示されている。

(1)	S	$\rightarrow$	$NP\ VP$
(2)	S	$\rightarrow$	VP

図 9: 文節間文法の例

(1)	PH	$\rightarrow$	NP
(2)	PH	$\rightarrow$	VP
(3)	NP	$\rightarrow$	N
(4)	NP	$\rightarrow$	$N\ P$
(5)	VP	$\rightarrow$	V
(6)	N	$\rightarrow$	$kore$
(7)	P	$\rightarrow$	o
(8)	V	$\rightarrow$	$kure$
(9)	V	$\rightarrow$	$okure$

図 10: 文節内文法の例

	NP	VP	\$	S
0	$s1$	$s2$		3
1		$s4$		
2			$r2$	
3			acc	
4			$r1$	

図 11: 文節間 LR 解析表の例

	e	*o*	*u*	*k*	*r*	$	*PH*	*NP*	*VP*	*N*	*P*	*V*
0							3					
1				*s*4				6		5		
2		*s*8		*s*7					10			9
3						*acc*						
4		*s*11										
5		*s*12				*r*3					13	
6						*r*1						
7			*s*14									
8				*s*15								
9						*r*5						
10						*r*2						
11					*s*16							
12						*r*7						
13						*r*4						
14					*s*17							
15			*s*18									
16	*s*19											
17	*s*20											
18					*s*21							
19		*r*6				*r*6						
20						*r*8						
21	*s*22											
22						*r*9						

文節行先表

文節カテゴリ	初期状態
NP	1
VP	2

図 12: 文節内 LR 解析表の例

4.1.2 2段階LR解析法に基づく文認識

2段階LR法に基づく文認識システムの構成を図13に示す。図13で、破線で囲まれた部分がHMM-LR文節認識部であり、右側の部分が文節間予測LRパーザとなっている。文認識は以下のように行なわれる。

1. 文節間予測LRパーザが文節間LR解析表を用いて、文法上で次にくる可能性のある文節カテゴリ（例えば、名詞句NP）を予測する。
2. 文節内予測LRパーザが、文節内LR解析表の文節行先表を参照して、予測された文節カテゴリに対する初期状態を決定する。
3. HMM-LR文節認識部を起動し、文節を認識し、結果を文節間予測LRパーザに返す。
4. ステップ1に戻る。

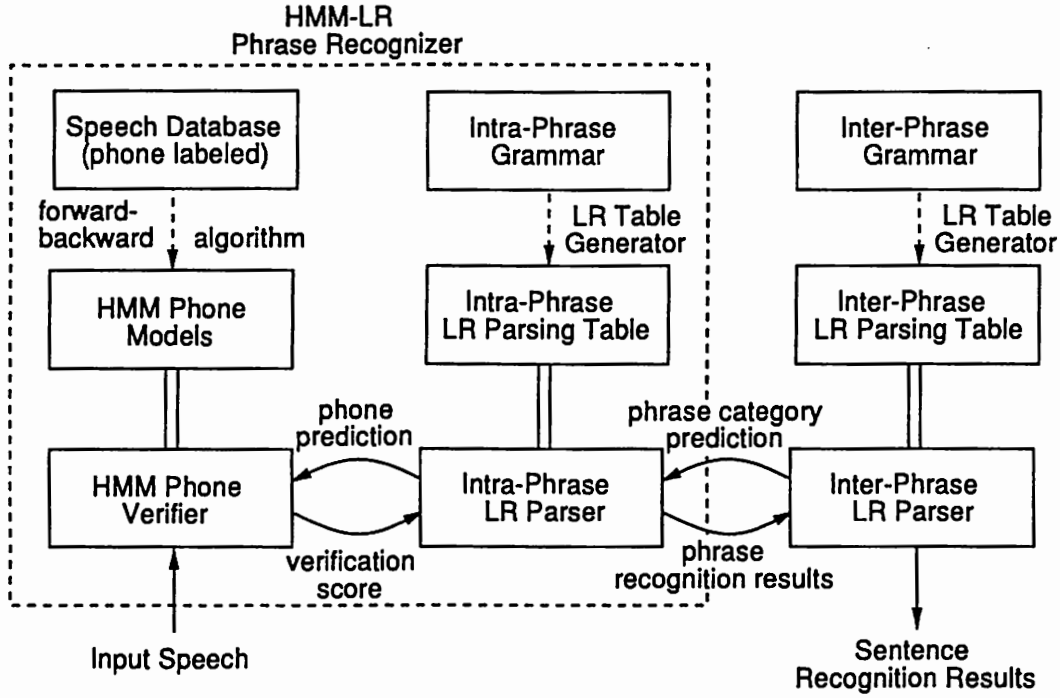


図13: 2段階LR法を用いた文認識システムの構成

4.1.3 文節内LR解析表の構成

文節内文法からどのようにして文節内LR解析表を構成すればよいかを、例（図14）を用いて説明する。

1. 図14の文法1が与えられたとする。文法1では、開始記号は *START* であり、*NP* と *VP* が文節カテゴリである。

2. 文法1の開始記号から文節カテゴリへの書き換え規則に対して、文法1に現れない新たな終端記号 (@*NP*, @*VP*) を用意し、その記号を書き換え規則の右辺の先頭に追加する。これにより、文法2を得る。

3. 通常のLR解析表構成アルゴリズムを用いて、文法2から解析表1を作成する。

4. 解析表1からステップ2で追加した終端記号を除いたものを解析表2-1とする。

5. 解析表1の初期状態（状態0）において、ステップ2で追加した終端記号のshift後の状態を調べ、解析表2-2を作成する。

以上の操作で作成された解析表2-1が動作表および行先表であり、解析表2-2が文節行先表となる。

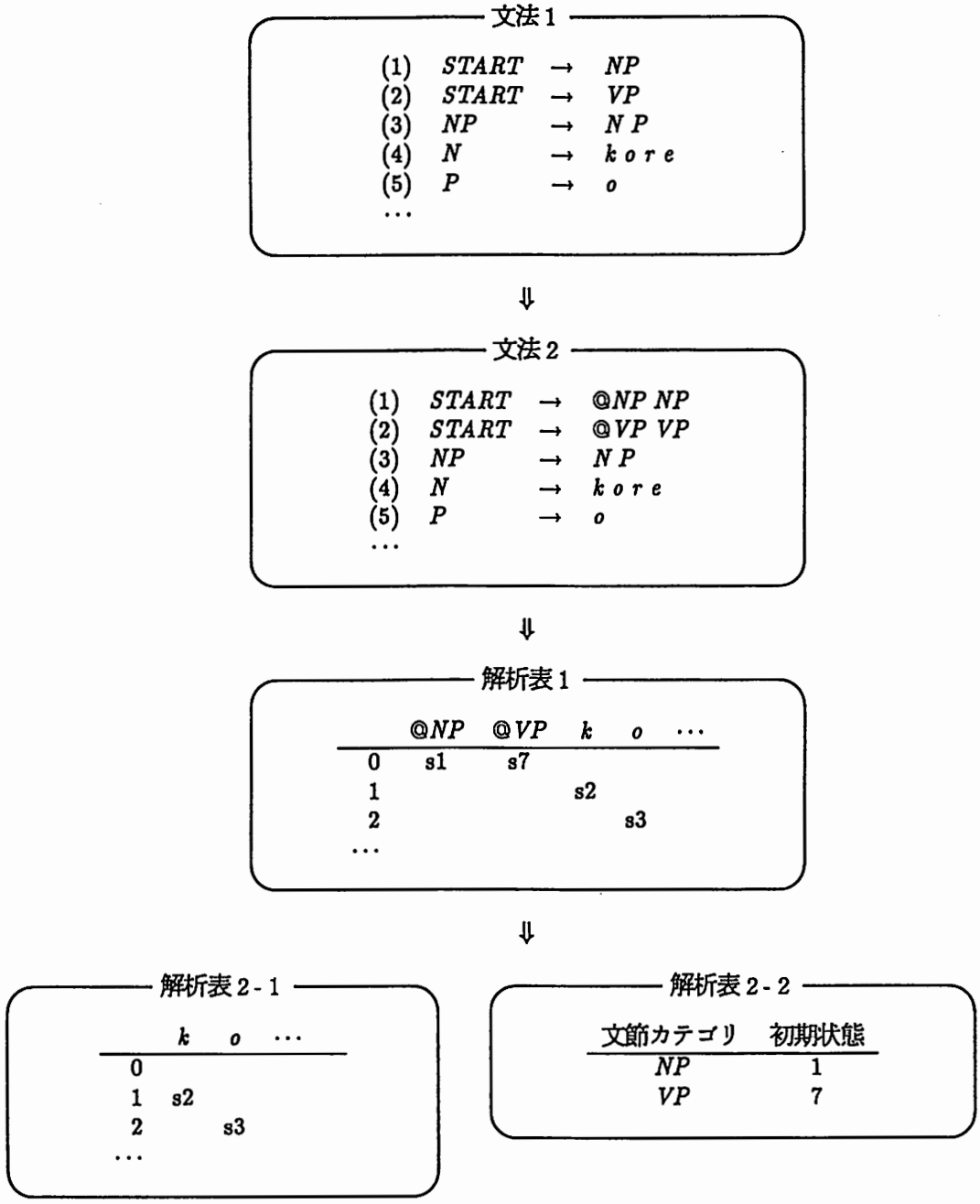


図 14: 文節内 LR 解析表の構成（例）

4.2 文節発声の場合の文認識による評価

2段階LR法に基づく文認識方式の有効性を調べるために、文節ごとに区切って発声された文データを用いて認識実験を行なった。

4.2.1 文法

2段階LR法の評価に用いられた文法の大きさおよび複雑さを表3に示す。なお、文節内文法は3.2.1で説明されているものと同じである。

表 3: 文節内文法, 文節間文法の大きさおよび複雑さ

	文節内文法	文節間文法
生成規則数	1,973 rules	471 rules
語彙数	744 words	133 phrase categories
パープレキシティ	3.57 / phone	65.5 / phrase

4.2.2 認識結果

次の2種類の認識実験を行なった。

(実験1) まず文節内文法のみを用いた HMM-LR 音声認識システムで、各文節発声データに対して、上位5個の認識候補 (以後、文節ラティスと呼ぶ) を求める。次に、文節間文法を用いて文節ラティスから文候補を作成する。

(実験2) 2段階LR法により、文節発声データから直接文候補を作成する。

実験結果を表4に示す。実験2 (2段階LR法を用いた場合) の方が、文節ラティスを経由した場合よりも文認識率で6%ほど向上しており、2段階LR法に基づく音声認識方式が文の認識において有効であることを示している。

表 4: 単語認識率および文認識率

順位	実験 1		実験 2	
	単語認識率	文認識率	単語認識率	文認識率
1	87.8	78.8	95.9	84.7
～ 2	-	85.4	-	90.5
～ 3	-	86.9	-	93.4
～ 4	-	86.9	-	94.9
～ 5	-	86.9	-	94.9

5 2段階 LR 法の改良

2段階 LR 法では、ある文節カテゴリが予測されたときに、その文節カテゴリに属するような音韻系列だけを文節内 LR パーザが生成する必要がある。これまでに説明してきた方法では、文節カテゴリごとに文節内 LR 構文解析表の初期状態を分割することにより、これを実現してきた。しかし、一般に認識途中では複数の文節カテゴリが予測されるために、文節内 LR パーザは複数の初期状態を保持する必要がある。また、異なる初期状態から同じ音韻を予測することがあり、この場合は音韻照合等の処理が重複して行なわれることになる。これらの問題を解決した改良アルゴリズムについて、以下で説明する。

5.1 LR-CRT アルゴリズム

ここでは、上位文法カテゴリへの到達可能性照合機構を持った LR 解析法 (LR Parsing with a Category Reachability Test, 以下 LR-CRT アルゴリズムと呼ぶ) [9] について説明する。

LR-CRT アルゴリズムは、ある文法カテゴリが与えられたとき、その文法カテゴリに属するような導出だけを求めるアルゴリズムである。HMM-LR 音声認識では、LR パーザを予測的に用いているので、このアルゴリズムを用いることにより、与えられた文法カテゴリに属する音韻系列だけを効率的に求めることができる。LR-CRT アルゴリズムは、通常の LR 法の拡張であり、LR 構文解析表の動作項が$< A, S >$ (A はパーザの動作、Sは到達可能な文法カテゴリのリスト) という形式をしている。LR-CRT アルゴリズムでは、ある文法カテゴリ Pが与えられたとき、Pが Sの要素であるときだけ動作 A を実行する。

図15に示す文法を、LR-CRT アルゴリズムで用いられる LR 構文解析表に変換した例を図16に示す。例えば状態0で記号 kには、$s3$ という動作と $\{S1, S2, S3\}$ というカテゴリの集合が付与されているが、これは $s3$ という動作を行なったあとでは $s1$ か $s2$ かあるいは $s3$ のいずれかに到達可能であることを示している。

(1)	PH	$\rightarrow$	$S1$
(2)	PH	$\rightarrow$	$S2$
(3)	PH	$\rightarrow$	$S3$
(4)	$S1$	$\rightarrow$	$k\ a$
(5)	$S2$	$\rightarrow$	$k\ i$
(6)	$S3$	$\rightarrow$	$k\ a\ i$

図 15: LR-CRT アルゴリズムで用いられる文法の例

	a	i	k	\$	PH	$S1$	$S2$	$S3$
0			$< s3, \{S1, S2, S3\} >$		5	1	2	3
1				$< r1, \{S1\} >$				
2				$< r2, \{S2\} >$				
3	$< s7, \{S1, S3\} >$	$< s6, \{S2\} >$						
4				$< r3, \{S3\} >$				
5				acc				
6				$< r5, \{S2\} >$				
7		$< s8, \{S3\} >$		$< r4, \{S1\} >$				
8				$< r6, \{S3\} >$				

図 16: 到達可能な文法カテゴリを持った LR 構文解析表

LR-CRT アルゴリズムを用いることにより、文節間 LR パーザで予測された文節カテゴリから次にくる音韻を効率的に予測することができる。いま、文節カテゴリとして p が予測されている場合、状態 s から予測される音韻の集合は次により求められる。

$$Prediction(s) = \{x | Action[s, x] = < \text{“}shift\text{”}, S > \ \& \ p \in S\} \qquad (2)$$

一般には複数の文節カテゴリが予測される。いま、Pを予測された文節カテゴリの集合とするとき、上式は次のように一般化される。

$$Prediction(s) = \{x | Action[s, x] = < \text{“}shift\text{”}, S > \ \& \ P \cap S \neq \phi\} \qquad (3)$$

5.2 改良された2段階LR法の評価

文節内LRパーザにLR-CRTアルゴリズムを用いることにより、2段階LR法に基づく文認識システムを改良した。4.2で説明したものと同じ音声データおよび文法を用いて評価を行なった結果を表5に示す。以前の結果と比べると、単語認識率で1.6%、文認識率で6.5%向上しており、改良されたシステムの有効性を示している。

表5: 改良された2段階LR法による単語認識率および文認識率

順位	単語認識率	文認識率
1	97.5	91.2
～2	-	93.4
～3	-	96.4
～4	-	97.1
～5	-	98.5

6 文連続音声認識

これまで述べてきた音声認識はすべて文節ごとに発声された音声を認識対象としていた。しかし、文節で区切って発声するというのは、話者に非常に大きな負担を与えることになる。文節発声という制約を取り除き、連続的に発声された文に対する連続音声認識について、次で説明する [10]。

7 方式概要

文連続音声認識の方式は HMM-LR 音声認識を用いている。ただし、文節認識のときと比べると、次のような改良がほどこされている。

1. 文節認識で使っていた HMM 音韻モデルの学習は、単語発声のデータを用いて行なわれた。しかし、連続音声中では音韻の変形が文節発声の場合よりも著しいので、HMM の学習データとして単語発声のデータに加え、文節発声，連続発声のデータを用いた。

2. HMM-LR 音声認識で文を認識するためには、文の文法が必要である。文の文法は 4.2で説明した文節間文法と文節内文法をマージすることにより作成した。

3. 文の場合は文節よりも構文的曖昧性が高いため、純粋に文法を用いるだけでは高い認識率を達成することは困難である。このため、確率的な言語モデルを併用した。

確率的言語モデルとして、確率文脈自由文法と生成規則の連鎖確率の 2 種類のものを用いた。以下で、各モデルについて説明する。

7.1 確率文脈自由文法

文脈自由文法では、$\alpha \rightarrow \beta$ という形の生成規則を用いることにより、言語の構造を階層的に記述する。このような文脈自由文法に対して、それを確率化した確率文脈自由文法 (probabilistic context-free grammar)[11] を次のように定義することができる。即ち、文脈自由文法中の各生成規則 $\alpha \rightarrow \beta$ に対して、左辺 α が右辺 β に書き換えられる条件付き確率 $P(\beta|\alpha)$ を付与するのである。但し、左辺に同じ非終端記号を持つ生成規則の確率を足し合わせると 1 になるように定義されなければならない。つまり、

$$\sum_{\beta} P(\beta|\alpha) = 1 \tag{4}$$

確率文脈自由文法を用いることにより、任意の導出木に対し、その導出木の生成確率を定義することができる。いま、文開始記号 S から文 x が、次のような導出により得られたとする。

$$S \stackrel{r_1}{\Longrightarrow} \gamma_1 \stackrel{r_2}{\Longrightarrow} \gamma_2 \stackrel{r_3}{\Longrightarrow} \cdots \stackrel{r_n}{\Longrightarrow} \gamma_n = x \tag{5}$$

このとき、この導出木 D の生成確率は、次の式によって計算できる。

$$P(D) = \prod_{i=1}^{n} P(r_i). \tag{6}$$

また、文 x の生成確率は、文 x に対するすべての導出木の生成確率の和で定義される。すなわち、

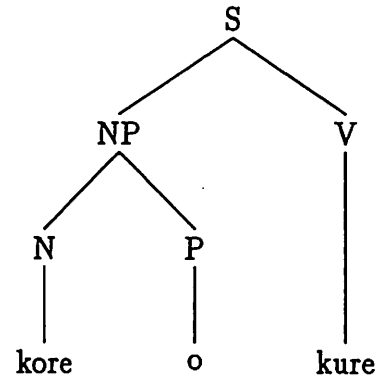
$$P(x) = \sum_{D} P(D). \tag{7}$$

従って、ある特定の導出木 D に対して、その相対確率 $P(D)/P(x)$ を導出木の尤度として用いることができる。

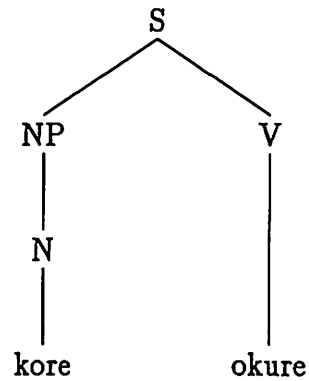
図 17に確率文脈自由文法の例を示す。図 17の文法によると、文 "$koreokure$" に対しては 2 通りの導出木が存在する。各導出木の生成確率の計算を図 18に示す。この場合、文 "$koreokure$" の生成確率は $0.224 + 0.084 = 0.308$ となる。

(1)	S	$\rightarrow$	$NP\ V$	0.7
(2)	S	$\rightarrow$	V	0.3
(3)	NP	$\rightarrow$	N	0.2
(4)	NP	$\rightarrow$	$N\ P$	0.8
(5)	N	$\rightarrow$	$kore$	1.0
(6)	P	$\rightarrow$	o	1.0
(7)	V	$\rightarrow$	$kure$	0.4
(8)	V	$\rightarrow$	$okure$	0.6

図 17: 確率文脈自由文法の例



$$
\begin{aligned}
&P(kore \cdot o \cdot kure) \\
&= P(NP \cdot V|S) \times P(kure|V) \\
&\quad \times P(N \cdot P|NP) \times P(o|P) \times P(kore|N) \\
&= 0.7 \times 0.4 \times 0.8 \times 1.0 \times 1.0 \\
&= 0.224
\end{aligned}
$$



$$
\begin{aligned}
&P(kore \cdot okure) \\
&= P(NP \cdot V|S) \times P(okure|V) \\
&\quad \times P(N|NP) \times P(kore|N) \\
&= 0.7 \times 0.6 \times 0.2 \times 1.0 \\
&= 0.084
\end{aligned}
$$

図 18: 導出木の生成確率

生成規則の適用確率は、大量の訓練用例文から推定する。文法が曖昧でないときは、各入力文に対して一意的に導出木が決定できるので、導出に用いられた各生成規則の適用回数を数え上げることにより推定できる。いま、全例文中で生成規則 $\alpha \to \beta$ が適用された回数を $n(\alpha,\beta)$ とすると、生成規則 $\alpha \to \beta$ の適用確率は、次式により推定できる。

$$P(\beta|\alpha) = \frac{n(\alpha,\beta)}{\sum_\gamma n(\alpha,\gamma)} \tag{8}$$

文法が曖昧であるときには、単純に生成規則の適用回数を数え上げるだけではうまくいかない。単純に数え上げるだけだと、曖昧な文ほど（導出木の数が多い文ほど）適用確率の決定に大きな影響を与えることになってしまう。直観的にいえば、例文 B に対する導出木 D に生成規則 r が n 回使われていた場合、D の相対確率 $P(D)/P(B)$ を考慮して、r が使われた真の回数は $(P(D)/P(B)) \times n$ としなければならない。

実際には、N 個の例文 $\{B^i\}$ から、以下のような繰り返し手続きにより適用確率を推定することができる [11]。なお、以下の繰り返し手続きは収束することが証明されている。

1. $P(\beta|\alpha)$ に初期値を代入する。但し、このとき $\sum_\beta P(\beta|\alpha) = 1$ とならなければならない。

2. 各例文 B^i に対するすべての導出木を求める。以下で、i 番目の例文の j 番目の導出木を D^i_j で表す。

3. 導出木 D^i_j の生成確率を次式により計算する。

$$P(D^i_j) = \prod_{r \in D^i_j} P(r) \tag{9}$$

4. 生成規則 $\alpha \to \beta$ が例文 B^i を生成するのに使われた回数 $C^i_\alpha(\beta)$ を次式により求める。

$$C^i_\alpha(\beta) = \sum_i \left(\frac{P(D^i_j)}{\sum_k P(D^i_k)} n^i_j(\alpha,\beta) \right) \tag{10}$$

ここで、$n^i_j(\alpha,\beta)$ は規則 $\alpha \to \beta$ が D^i_j の導出に使われた回数を表している。

5. 左辺に同じ非終端記号 α を持つ生成規則の総回数が 1 となるように、回数を正規化する。

$$f_\alpha(\beta) = \frac{\sum_i C^i_\alpha(\beta)}{\sum_i \sum_\gamma C^i_\alpha(\gamma)} \tag{11}$$

6. $P(\beta|\alpha)$ を $f_\alpha(\beta)$ で置き換え、ステップ 2 から繰り返す。

確率文脈自由文法から、確率付きの構文解析表を構成する方法が知られており [12]、確率付きの解析表を用いることにより効率的に文（あるいは部分文）の生成確率を計算することができる。確率文脈自由文法を使うだけだと、生成規則の適用が起こるまでは文（部分文）の生成確率を計算することができないが、確率付きの解析表を用いることにより、記号が 1 つ shift されるごとに生成確率を計算することができる。

図 17の確率文脈自由文法からは、図 19に示す構文解析表が得られる。この動作表を用いた導出木の生成確率計算は次のようになり（$P(a|s)$ は状態 s で動作 a が起こる確率を表している)、当然のことながら図 18での計算結果と一致している。

$$
\begin{aligned}
& P(kore \cdot o \cdot kure) \\
&= P(s4|0) \times P(s8|4) \times P(s15|8) \times P(s18|15) \times P(r5|18) \\
&\quad \times P(s13|6) \times P(r6|13) \times P(r4|12) \times P(s11|5) \times P(s9|11) \\
&\quad \times P(s16|9) \times P(s19|16) \times P(r7|19) \times P(r1|10) \times P(acc|1) \\
&= 0.82 \times 0.85 \times 1.0 \times 1.0 \times 1.0 \\
&\quad \times 0.8 \times 1.0 \times 1.0 \times 0.4 \times 1.0 \\
&\quad \times 1.0 \times 1.0 \times 1.0 \times 1.0 \times 1.0 \\
&= 0.224
\end{aligned}
$$

7.2 生成規則の連鎖確率

生成規則の連鎖確率モデルは、ある生成規則のあとにはどの生成規則が使われやすいかという生成規則の適用順序に基づいた確率的言語モデルである。文脈自由文法では、文脈に依存せずに生成規則が適用されるが、このモデルは文脈による生成規則の適用頻度を考慮したモデルとなっており、文脈依存性を確率の形で持っているといえる。

現在、HMM-LR 音声認識システムでは、生成規則の bigram モデルを用いている。このモデルでは、文開始記号 S から文 x が、

$$S \overset{r_1}{\Longrightarrow} \gamma_1 \overset{r_2}{\Longrightarrow} \gamma_2 \overset{r_3}{\Longrightarrow} \cdots \overset{r_n}{\Longrightarrow} \gamma_n = x \tag{12}$$

という導出で得られたとき、この導出木 Dの生成確率を、

$$P(D) = P(r_1, \ldots, r_n) = P(r_1 \mid \#)P(r_2 \mid r_1) \prod_{k=3}^{n} P(r_k \mid r_{k-1})P(\# \mid r_n) \tag{13}$$

によって計算する。実際には、LR パーザは bottom-up に解析を進めるので、HMM-LR 音声認識システムは $P(r_n, \ldots, r_1)$ を尤度として用いている。

	e	*o*	*u*	*k*	*r*	$	*S*	*NP*	*V*	*P*	*N*
0		(*s*3, 0.18)		(*s*4, 0.82)			1	5	2		6
1						(*acc*, 1.0)					
2						(*r*2, 1.0)					
3				(*s*7, 1.0)							
4		(*s*8, 0.85)	(*s*9, 0.15)								
5		(*s*3, 0.6)		(*s*11, 0.4)					10		
6		(*s*13, 0.8)		(*r*3, 0.2)						12	
		(*r*3, 0.2)									
7			(*s*14, 1.0)								
8					(*s*15, 1.0)						
9					(*s*16, 1.0)						
10						(*r*1, 1.0)					
11			(*s*9, 1.0)								
12		(*r*4, 1.0)		(*r*4, 1.0)							
13		(*r*6, 1.0)		(*r*6, 1.0)							
14					(*s*17, 1.0)						
15	(*s*18, 1.0)										
16	(*s*19, 1.0)										
17	(*s*20, 1.0)										
18		(*r*5, 1.0)		(*r*5, 1.0)							
19						(*r*7, 1.0)					
20						(*r*8, 1.0)					

図 19: 確率付き LR 構文解析表

8 評価

モデル会話A，B，1～5の連続発声データ（全１３７文）を用いて、文連続音声認識実験を行なった結果を、表6に示す。まずHMMの学習データであるが、単語発声のデータのみを用いたときよりも、文節発声，連続発声のデータを付け加えたときの方が高い認識率を達成することができた。また、文法のみを用いるよりも確率的言語モデルを併用した方が認識率が高く、確率文脈自由文法よりも生成規則の連鎖確率モデルの方が優れていることが分かる。最終的には、生成規則の連鎖確率モデルを用いたときに、文認識率83.9%、5位までの累積認識率86.1%を達成することができた。

表 6: 文連続音声認識実験結果

HMM 学習データ	言語モデル	単語認識率					文認識率	
		Correct	*Subs*	*Dels*	*Ins*	*Accuracy*	*Correct*	*Within top* 5
単語発声	文法のみ	55.2	12.9	31.9	11.9	43.3	48.9	56.9
	確率文法	56.6	7.8	35.7	5.1	51.4	59.9	66.4
	規則の連鎖確率	77.8	5.6	16.6	1.4	76.4	78.1	81.0
単語発声＋文節発声	文法のみ	66.7	12.5	20.8	6.8	59.9	55.5	64.2
	確率文法	69.1	9.2	21.8	3.3	65.7	64.2	72.3
	規則の連鎖確率	83.1	2.9	14.0	0.1	83.0	83.2	86.9
単語発声＋文節発声＋連続発声	文法のみ	70.8	13.5	15.7	6.9	63.9	57.7	65.0
	確率文法	74.1	10.5	15.4	4.1	70.0	66.4	71.5
	規則の連鎖確率	81.0	2.9	16.1	0.0	81.0	83.9	86.1

Correct：正解率，*Subs*：置換エラー率，*Dels*：脱落エラー率，*Ins*：挿入エラー率，
$Accuracy = 100 - Subs - Dels - Ins$

参考文献

[1] Hopcroft, J. E., Ullman, J. D.: Introduction to Automata Theory, Languages, and Computation, Addison-Wesley (1979).
野崎 昭弘 (他訳) オートマトン 言語理論 計算論 (I,II), サイエンス社 (1984).

[2] Aho, A. V., Sethi, R., Ullman, J. D. : Compilers — Principles, Techniques, and Tools, Addison-Wesley (1986).
原田 賢一 (訳) : コンパイラ (I,II), サイエンス社 (1990).

[3] Tomita, M. (ed.) : Generalized LR Parsing, Kluwer Academic Publishers (1991).

[4] 田中 穂積, Suresh, K. G. : YAGLR 法 : Yet Another Generalized LR Parser, 情報処理学会自然言語処理研究会, 83-11 (1991).

[5] 北 研二, 川端 豪, 斎藤 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. 472-480 (1990).

[6] 花沢 利行, 北 研二, 中村 哲, 川端 豪, 鹿野 清宏 : HMM-LR 音声認識システムの性能評価, 日本音響学会誌, Vol. 46, No. 10, pp. 817-823 (1990).

[7] 北 研二, 江原 暉将, 森元 逞 : HMM-LR 音声認識の大語彙への適用, 情報処理学会第 4 2 回全国大会, pp. 2-110-2-111 (1991).

[8] Kita, K., Takezawa, T., Morimoto, T.: Continuous Speech Recognition Using Two-Level LR Parsing, IEICE Transactions, Vol. E74, No. 7, pp. 1806-1810 (1991).

[9] 北 研二, 森元 逞, 嵯峨山 茂樹 : 上位文法カテゴリへの到達可能性照合機構を備えた LR 解析法とその音声認識への応用, 日本音響学会秋季講演論文集, 2-P-17, pp. 171-172 (1991).

[10] 北 研二, 森元 逞, 大倉 計美, 嵯峨山 茂樹 :「統計的言語情報を用いた HMM-LR 文章発声音声認識の評価」, 情報処理学会ヒューマンインタフェース研究会 44-4 or 電子情報通信学会音声研究会 SP92-53 (1992).

[11] Fujisaki, T., Jelinek, F., Cocke, J., Black, E., Nishino, T.: A Probabilistic Parsing Method for Sentence Disambiguation, In *Current Issues in Parsing Technology*, Tomita, M. (Ed.), pp. 139-152, Kluwer Academic Publishers (1991).

[12] Wright, J. H. : LR Parsing of Probabilistic Grammars with Input Uncertainty for Speech Recognition, Computer Speech and Language, No. 4, pp. 297-323 (1990).