音声言語情報処理 14－17
(1996. 12. 13)

効率的な仮説のマージ機能を持つLRパーザ制御による音声認識

山田智一　松永昭一　嵯峨山茂樹

NTTヒューマンインタフェース研究所

〒239 神奈川県横須賀市光の丘1-1
{tomokazu,mat,saga}@nttspch.hil.ntt.co.jp

あらまし　文脈自由文法(CFG)による言語制約下で、環境依存音素モデルを用いる連続音声認識アルゴリズムを提案する。一般化LRパーザを用いて、パスのマージを考慮して、有限状態ネットワーク(FSN)をCFGから動的に生成し、時間同期に処理を行う。音響モデルは、単語間、単語内の環境依存性を考慮する。本稿では、アルゴリズムと処理機構について述べ、さらに音素同期に処理を行うHMM-LR法との比較実験結果について述べる。HMM-LR法と比べると、本手法は、認識率を落とさずに処理時間を削減することができた。

キーワード　音声認識　HMM　LRパーザ　one-passサーチ　有限状態オートマトン

Continuous Speech Recognition Using LR Parsing With Effective Hypotheses Merging Mechanism

Tomokazu YAMADA　Shoichi MATSUNAGA　Shigeki SAGAYAMA

NTT Human Interface Labolatories

1-1, Hikarinooka, Yokosuka-Shi, Kanagawa, 239 Japan

Abstract　This paper describes a Viterbi search algorithm for continuous speech recognition using context-dependent phone models under the constraint defined by a context-free grammar (CFG). It is based on a frame synchronous LR parser which dynamically generates a finite state network (FSN) from the CFG with an efficient path merging mechanism. Full context-dependency (intra- and inter-word context) is taken into account. This paper first describes the algorithm and the processing mechanism, then compares the experimental results of our algorithm and the conventional tree-based HMM-LR speech recognition algorithm which uses HMMs and an LR parser in phone-synchronous processing. The experiments show that our algorithm runs faster than the HMM-LR algorithm with an equivalent recognition accuracy.

key words　Speech Recognition, HMM, LR Parser, One-pass Search, Finite State Automaton

1 はじめに

近年の大語彙連続音声認識では、大量のテキストデータが利用できる場合、統計的言語モデルを作成し、これをオートマトン制御によるOne-Passアルゴリズム[1]に基づいた手法で利用することが多い。One-Passアルゴリズムが持つ、アルゴリズム自体が簡潔で実装が容易なこと、時間同期に処理を行なうので実時間処理に向いていること、などの利点もあって、実際、良好な性能を示している。

一方で、小・中語彙連続音声認識の場面も多く考えられているが、このようなタスクでは必ずしも統計的言語モデルを作成するのに十分なテキストデータが得られないことが多い。また、統計的言語モデルを用いた場合に比べ、探索空間をかなり絞り込むことが可能であり、処理後に意味解析等の言語処理を行う場合には整合性が取りやすくなることが考えられる。このようなことから、可能であれば文法によって言語的制約を設定したいという要求は強い。

実際に文法を用いた種々の音声認識手法が提案されているが、 比較的記述が容易な文脈自由文法(CFG)を用いた手法が多い[2]-[9]。CFGの具体的な利用方法は様々であるが、よく知られたものの一つに、LR構文解析を用いて音素同期に処理を行うHMM-LR法[6]がある。あらかじめ作成されたLR表を用いることにより、音声の始端方向からの同一音素系列の仮説をひとまとめにして処理することができるので、音声認識処理に都合がよい。しかし、音素同期に処理を進めるため、認識の過程で各仮説が評価に用いている音声区間長が異なることから、時間長の正規化を考慮したビーム探索を行う必要があること、仮説のマージを考えると、その条件の設定が困難で、結局時間同期的な処理が必要になってしまうこと、などの問題点があった。そのため、LR構文解析を用いて、フレーム同期で処理する探索手法がいくつか提案されている[7][8][9]。これらの手法では、CFGからあらかじめFSNを作成すると、文法によっては非常に大きなネットワークを生成してしまうので、認識処理の過程で見込みのありそうな仮説に関してのみ動的にネットワークを生成することにより、メモリを効率よく使うことが検討されている。また、フレーム同期に処理されるので、 音素同期のHMM-LR法のように、時間正規化やA*探索のようなヒューリスティック関数の設定等がなくてもビーム探索が可能であり、LR構文解析上、マージ可能な仮説を見つけることが比較的容易であるので(2.1節参照)、これを考慮してネットワークを生成することもできる。

本稿では、言語的制約としてCFGを利用可能とし、これからLR構文解析を用いて有限状態ネットワーク(FSN)を動的に生成し、One-Pass Viterbiアルゴリズムに基づいてフレーム同期に処理する探索手法を提案する。基本的な枠組みは[7][8][9]と同様であるが、さらに以下のような機能を加えることにより、処理効率・性能の改善を図った。

- 音響モデルとして環境依存音素モデルを用い、単語内だけでなく単語間についても環境依存性を考慮する
- LR構文解析の過程で生成されるスタックの内容に応じたノードを効率的に利用し、FSNのノードとする
- Delayed Arc Evaluationと呼ぶ方法で、環境独立音素によるFSNと同等のマージ効率を維持しながら、環境依存音素モデルからなるFSNを動的に生成する

まず本手法の詳細について述べ、次にHMM-LR法との比較実験の結果について述べる。実験結果から本手法の有効性を示す。

2 探索アルゴリズム

2.1 LR構文解析に基づくパスのマージ

時間同期処理の場合には、HMM-LR法のように各仮説が評価に用いている音声区間長が異なることはないので、比較的容易に仮説のマージが可能である。マージする仮説は、LR構文解析に基づいて決定する。すなわち、異なる2つ(以上)の仮説の、LR構文解析用のスタックに積まれた状態系列が同一であるものをマージする。これは、スタックの内容が同一であれば、それ以降の解析結果も同一となることに基づいている。簡単な日本語の文節文法(図1)と、それを元に展開された仮説の例(図2)を使って、仮説のマージ方法について説明する。図1の文法はあらかじめLRテーブルに展開されている。スタックに状態0だけが積まれた仮説から始まり、テーブル上の可能な動作によって予測的に仮説を伸ばしていく。このとき、動作に応じて状態がpushまたはpopされ、各仮説のスタックの内容が変化する。

(0) **S** → - **phrase** -
(1) **phrase** → **town particle**
(2) **town** → k o o f u
(3) **town** → k o o b e
(4) **town** → k o g a n e i
(5) **particle** → k a r a

図1 日本語文節文法の例

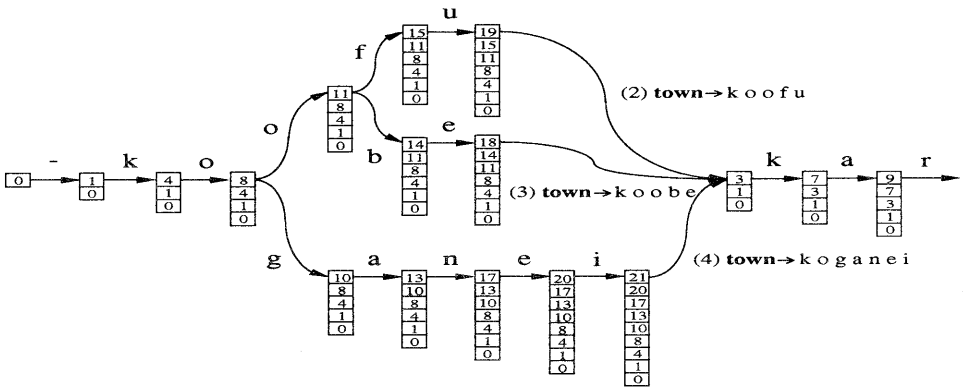

図2　LR構文解析に基づいた仮説のマージの例

図2に示された数字は、 各仮説に対するスタックに積まれた状態番号を示している。 この文法では **town** として3つの単語が存在するので、一旦3つのパスに分かれるが、生成規則(2), (3), (4)によるreduce動作により、各パスともスタックに積まれた状態が [0, 1, 3] というLR上の同一の状態系列に達する。スタックの状態が同じであれば、それ以降の解析動作は等しくなる、すなわち予測される音素系列は等しくなるので、これら3つのパスをマージすることができる。

2.2 Delayed Arc Evaluationに基づく動的ネットワーク展開

音響モデルとしては環境依存音素モデルを用いる。これをLRパーザで扱うには、以下の2つの方法が考えられる。

(a) CFGの終端記号を環境依存音素モデルを表す記号で記述し、LRパーザのshift動作で、直接環境依存音素を予測する

(b) CFGの終端記号は環境独立音素とし、認識時に、LRパーザの予測した環境独立音素の系列から、適切な環境依存音素モデルを決定する

(a)の場合にも種々の実現方法が考えられるが、これを単純に行うと、(b)の場合に比べ、仮説のマージポイントが分散してしまうと考えられるので、(b)の方法に基づいて仮説を生成することにした。

図3に示したネットワーク(FSN)を用いて具体的に説明する。これは、ある文章を受理するネットワークが途中まで生成された例である。ここでは環境依存モデルとしてtriphoneを考える。まず、環境独立音素を終端記号とするCFGを用いて、LRパーザが用いる各スタックの内容に1対1で対応するノード(図中の○)を作成する。1対1の意味するところは、あるノードから新たに作成しようとしたノードのスタックの内容が、すでに別のノードから作成されていたノードのスタックの内容と同一であった場合に、新たにノードを生成せずに、単にそのノードへのパスを設定することである。これによりパスのマージが行なわれることになる。shift 動作で生成されたノードでは、前のノードとの間に(shiftされた終端記号である) 環境独立音素が仮定される (図中の点線部分のパス)。reduce動作の場合には設定される環境独立音素はない(null遷移 ϕ)。

このようにLRパーザを用いて新たに作成されたパスと、それが接続されたノードに対し、ノードはそのまま共用し、必要なノード間にアークを張り付けることで、Viterbi探索に用いるネットワークを生成する。具体的には元になったノードに入ってくるアーク各々に対し、そこに設定されている環境依存音素の中心、 及び右側の環境を調べる。これらとshift動作で予測された環境独立音素とから、アークに設定すべき環境依存音素モデルを決定する(図中、実線部分)。図中 N_p と N_n で説明すると、元になったノード N_p に入ってくるアークに設定されている環境依存音素モデル ${}_o o_b$ の中心音素 o と右側の環境の音素 b、及び N_p-N_n 間のパスで予測された環境独立音素 e の3つの音素から得られる ${}_o b_e$ という環境依存音素モデルを、ノード間に張るアークに設定する。

元になるノードに入ってくるアークがnull遷移の場合には、再帰的にその元のノードを遡り、そのノードとの間に直接アークを設定する。(このため、LRパーザによるnull遷移パスの設定時には、それをそのままnull遷移のアークとしておく。) LRパーザで予測された環境独立音素が、1ステップずつ

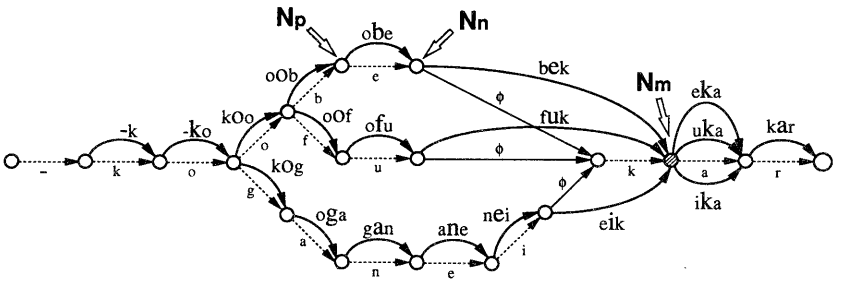

図3　環境依存音素モデルに基づくネットワークの生成例

遅れて中心音素に設定される形になるので、この手法を Delayed Arc Evaluation (DAE) と呼ぶ。この方法では、図中 N_m のような、本来マージすべきでないマージノードが生成される。この部分については、対応する入力アークと出力アーク間で別のノードを作成するか、マージノードの尤度を用いずに対応するアーク間で正確な計算を行うようにすればよい(そのまま近似計算として利用することも考えられる)。

LR パーザを用いた DAE に基づく動的ネットワーク展開をまとめると、以下のようになる。

A : active ノードの集合
S : shift 動作を行なうノードの集合
R : reduce 動作を行なうノードの集合

ReduceNode(r, n) :
規則 r を用いてノード n から reduce 動作によって生成された新たなノード

ShiftNode(p, s, n) :
ノード n に対し shift 動作で音素 p、状態 s を push して生成された新たなノード

ExistNode(n) :
もしすでに n と同一のスタック内容を持つノードが存在したら、そのノードを返す

SourceNode(a) :
アーク a のソースノードを返す

ExistConnection(n, m, p) :
もしすでに終端記号 p のノード n から m への接続が存在したら TRUE、存在しなければ FALSE

MakeLRConnection(n, m, p) :
終端記号 p で、ノード n から m へ接続する

Model(a) :
アーク a の環境依存音素モデルを返す

ExistArc(n, m, q) :
もしすでに間起用依存音素モデル q で、ノード n から m へのアークが存在したら TRUE、存在しなければ FALSE

GetModel(q, p) :
環境独立音素 p と triphone q の中心及び右側音素の組で構成される環境依存音素モデルを返す

■ DAE に基づく動的 FSN 生成アルゴリズム

$N \leftarrow A$, $S \leftarrow \phi$, $R \leftarrow \phi$

$A \leftarrow \phi$

while $N \neq \phi$ **do**
  for each node $n \in N$ **do**
    for each possible LR action x for n **do**
      if x is reduce action **then**
        $R \leftarrow R \cup < r, n >$
        where r is applied rule number
      else if x is shift action **then**
        $S \leftarrow S \cup < p, s, n >$
        where p is the shifted CI-phone,
        s is the LR state number
      end
    end
    remove n from N
    $A \leftarrow A \cup \{n\}$
  end
  if $R \neq \phi$ **then**
    call MakeReduceNodes
  end
end
call MakeShiftNodes

■ MakeReduceNodes
(reduce 動作の対象ノード群の処理)

```
N ← φ
for each item < r, n > ∈ R do
    n' ← ReduceNode(r, n)
    m ← ExistNode(n')
    if m ≠ φ then
      N ← N ∪ {m}
      if not ExistConnection(n, m, φ) then
          MakeLRConnection(n, m, φ)
      end
    else
      MakeLRConnection(n, n', φ)
      N ← N ∪ {n'}
    end
end
```

■ **MakeShiftNodes**
(shift 動作の対象ノード群の処理)

```
A ← φ
for each item < p, s, n > ∈ S do
    n' ← ShiftNode(p, s, n)
    m ← ExistNode(n')
    if m ≠ φ then
      n' ← m
    end
    call SetArcs with item < n, n', p >
end
```

■ **SetArcs**(n, n', p)
(n、n' 間に、予測された音素 p に対応するアーク群を設定する)

```
for each input-arc a of n do
    if a ≠ null-transition then
      q ← Model(a)
      q' ← GetModel(q, p)
      if not ExistArc(n, n', q') then
          set arc from node n to n'
          with context-dependent model q'
      end
    else
      n'' ← SourceNode(a)
      call SetArcs with item < n, n'', p >
    end
end
A ← A ∪ n'
```

2.3 認識アルゴリズム

認識アルゴリズム全体をまとめると、以下のようになる。本手法では、あらかじめ最初の数ノードを生成してからでないと、環境依存音素が設定できない。以下に FSN pre-generation とあるのは、このための処理である。

- FSN pre-generation 及び変数等の初期化
- 各フレームに対し (a) から (c) を繰り返す
  - (a) DAE による動的ネットワーク展開
  - (b) FSN に基づく one-pass Viterbi 探索
  - (c) active ノードの設定 (仮説の枝刈り)
- バックトレース

3 評価実験

本手法の有効性を確認するため、HMM-LR 法との比較実験を行った。両手法のビーム探索の手法が異なるので、直接ビーム数などにより比較することはできない。そこで、処理時間と認識率との関係による比較を行った。

音響モデルには HMnet に基づく不特定話者用の環境依存音素モデル [10] を用いた。実験タスクとしては、

(A) 電子協 100 都市名の前後に不要語を付与した文章、4 話者各 88 発声

(B) 国際会議問い合わせ (文節)、1 話者 280 発声

の 2 つを用いた。(A) では、マージの効果とマージノードでの環境依存音素の設定が正しく行われていることを確認するため、不要語の音響モデル等を用いるのではなく、文法中に不要語も音素系列で記して対応した。文法で受け付ける文章は、

sentence → pre_garbage cityname post_garbage

のような形である。(B) には、このタスク用の文節文法 (単語数 1078、文法数 2679) を用いた。

結果を図 4、図 5に示す。(A) の実験では、単語正解率がほぼ飽和したところで見ると、本手法の方が、同一の処理時間で高い認識性能を示した。これは仮説のマージを行ったことが有効に機能していることを示していると考えられる。一方、(B) の実験では、その差はあまり見られなかった。これは、文節発声の場合、文節末のわずかな助詞や活用語尾程度しかマージを行うことができなかったためではな

いかと考えられる。また、HMM-LR法として用いた認識系は、すでにかなり処理の高速化が図られたシステムであったことも理由として考えられる。現時点ではまだ、ネットワークの動的生成部の負荷が、全体的な処理の中で無視できない割合を占めている。これは、すでに生成されているノードに後から接続されるパス対し、適切な環境依存音素を設定する等の、再帰的なノードのチェックが必要であることが大きく影響している。今後は、この点を含めて、さらに処理の高速化を検討していく。

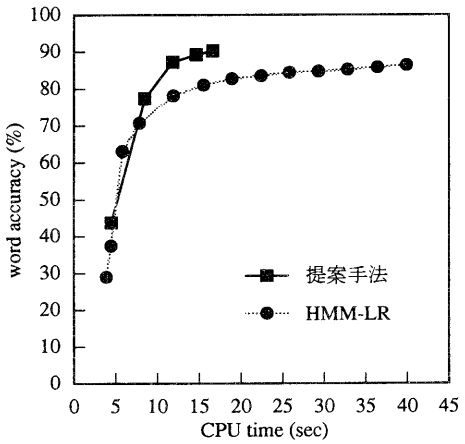

図4　不要語を含む100都市名認識(文章)における単語正解率と処理時間の関係

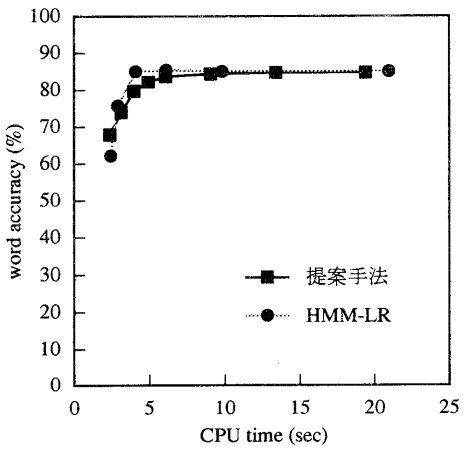

図5 国際会議問い合わせ(文節)における単語正解率と処理時間の関係

4　むすび

DAEによる動的ネットワーク展開手法を用いた、LRパーザ制御によるViterbi探索アルゴリズムを提案した。不要語を含む電子協100都市名の認識実験では、従来のHMM-LR法に比べ、単語認識率85%を実現するのに、処理時間を約70%削減することができた。今後は、文法の種類や規模によらず、高速な処理が行なえるよう、改善を図っていく。

謝辞

音響モデルを提供してもらったNTTヒューマンインタフェース研究所の高橋敏氏に感謝致します。

参考文献

[1] H. Ney: "The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition," IEEE Trans. Acoustics, Speech and Signal Processing, **ASSP-32**, 2, pp. 263–271(1984).

[2] H. Ney: "Dynamic Programming Speech Recognition using a Context-Free Grammar," ICASSP, pp. 69–72(1987).

[3] Y. L. Chow, M. O. Dunham, O. A. Kimball, M. A. Krasner, G. F. Kubala, J. Makhoul, P. J. Price, S. Roucos, and R. M. Schwartz: "BYBLOS: The BBN Continuous Speech Recognition System," ICASSP, pp. 89–92(1987).

[4] S. Nakagawa: "Spoken Sentence Recognition by Time-Synchronous Parsing Algorithm of Context-Free Grammar," ICASSP, pp. 829–832(1987).

[5] X. Huang, A. Acero, F. Alleva, M. Hwang, L. Jiang, and M. Mahajan: "Microsoft Windows Highly Intelligent Speech Recognizer: Whisper," ICASSP, pp. 93–96(1995).

[6] 北, 川端, 斉藤: "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, (1990).

[7] K. Itou, S. Hayamizu, and H. Tanaka: "Continuous Speech Recognition by Context Dependent Phonetic HMM and an Efficient Algorithm for Finding N-best Sentence Hypotheses," ICASSP, 10.6, pp. 21–24(1992).

[8] K. Kita, Y. Yano, and T. Morimoto: "One-Pass Continuous Speech Recognition Directed by Generalized LR Parsing," ICSLP, 1.4, pp. 13–16(1994).

[9] T. Shimizu, S. Monzen, H. Singer, and S. Matsunaga: "Time-Synchronous Continuous Speech Recognizer Driven by a Context-Free Grammar," ICASSP, RP02.07, pp. 584–587(1995).

[10] 高橋, 嵯峨山: "4階層の共有構造を持つ音素環境依存HMMの検討," 音学講論(秋), 3-8-3(1994).