漸次的に単語部分木を出力する音声認識システム *

森　元　　逞 **
高　橋　伸　弥 **
吉　村　賢　治 **
乙　武　北　斗 **

Incremental Speech Recognition System Outputting Partial Word Sub-tree

Tsuyoshi MORIMOTO**, Shin'ya TAKAHASHI**, Kenji YOSHIMURA**
and Hokuto OTOTAKE**

In human dialogue, a hearer can recognize input speech incrementally, and, can response back with acknowledge/negative signs such as "ya/um?" or nodding while a speaker is still speaking. For realizing such human-like speech dialogue system, we have developed a speech recognition system which can recognize incrementally and output partial word sub-tree.

Key Words : Speech Recognition, Incremental Speech Recognition, Spoken Dialogue

１．はじめに

現在まで多くの音声認識システムが開発され，また実用化されているシステムも多いが，それらのほとんどにおいて認識結果が出力されるのは，発話が終了した時点以降になる．これは発話の入力に対し，内部的な認識処理は同時並行的に行うものの，途中では極めて多数の候補が生成されるため出力は行わず，発話終了時にそれらの中から最良（最尤）スコアのものを確定し，その単語列を認識結果として出力するようにしているためである．その結果，発話全体を通して最も確からしい認識結果を出力することができるが，後段の言語処理にとっては，発話終了時まで処理を開始することができないことになる．

一方，人と人との対話においては，聞き手は聞き取った発話を順次認識し，理解している．これにより聞き手は，発話の途中においても適切な「あいづち」や「うなづき」などをバックチャネルとして相手（話し手）に返すことができる．また話し手は，聞き手からのこのようなバックチャネルによって相手が本当に聞き取れたか，理解できたかを確認しながら，話しを展開していく．このように，人と人との会話では順次認識を行うことが，確実な情報伝達や，いきいきとした対話の実現に極めて重要となっている．

我々は，このような人に近い音声認識，言語理解システムを構築することを目標としている．そこで，まず第一段階として，音声認識の途中において，その時点までに認識された単語を漸次出力するシステムを開発した．

これまでも，認識の途中段階で結果を出力する方式がいくつか提案されている[1]-[3]．しかしこれらはいずれも音声認識結果として１つに絞り込めた（またはその可能性が非常に高い）部分を確定部分として早期に出力しようというものである．

一方，我々の目標は，前述したように，後続の言語処理も含めて漸次理解を行うことができるシステムの開発である．このようなシステムでは，候補の絞り込みはむしろ言語処理で実現した方が良いと思われる．なぜなら，言語処理の方が意味や文脈などの知識を援用してより適切な絞り込みができ，また言語的な判断も加えて適

* 平成 25 年 5 月 31 日受付
** 電子情報工学科

切なバックチャネルを返すなどの処理が実現できるためである．そこで，音声認識では認識結果が確定しなくても，認識できた候補を漸次単語木の形で出力する方法とする．また一般に連続音声では，単語間の境界には曖昧性があることから，可能性のある境界を少しずつずらしながら認識を行ってそれらすべてを候補としているため，時間の経過とともに同じ単語列（パス）ではあるが各単語境界が少しずつ異なる多数の候補が生成される．これらをすべて出力すると後続の言語処理で扱いきれなくなる．この問題を解決するため，パス情報を２階層で管理する．１段目では単語境界が異なれば別パスとして管理するが，２段目では単語境界が異なっても同じ単語列として管理する．また適当なタイミングで１段目の中から最良なものを選んで，スコア付きで出力する．なおこれらは局所的に最良と判断されたものであるから，後刻，さらにスコアの良いものが認識されることがあるが，その場合は再出力するようにする．

本認識システムのベースとしては，HTKのHVite[4]を用い，そこに上記のような漸次出力機能を組み込んだ．また言語モデルとしては，我々が以前開発した，DPマッチングにより例文からFSA言語モデルを自動生成するシステム[6]を用いて作成した文法を用いた．

以下では，まず音声認識の原理を簡単に説明する．次に我々が実現した漸次音声認識の方式について述べる．また，300文程度の旅行会話文を対象とした動作例を報告し，本手法の有効性を議論する．

２．音声認識の原理

最初に一般的な音声認識の原理について簡単に説明する．システムには，①音響的な特徴を定義したHMM，②そのシステムで認識対象となる単語を定義した単語辞書，③単語間の接続制約等を定義した言語モデル，の3種の情報が用意される．

HMMは音素を単位として定義されるが，各HMMはいくつかの状態とそれら状態間の遷移で構成されている(Fig.1)．先頭および最後尾の状態はそれぞれ「開始状態」「終了状態」である．開始状態から次の状態には直ちに遷移するが，それ以降の状態遷移は１フレームごとに行われる．またこれらの遷移には，遷移確率と記号の出力確率[1)]が定義されている．また状態間の遷移として，ある状態から自分自身への遷移（自己遷移）も定義されている．

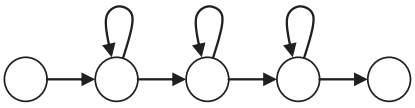

Fig.1　HMM

状態と状態間の遷移から成る．先頭および最後尾の状態はそれぞれ「開始状態」「終了状態」である．この図では，状態間の遷移確率と，遷移にともなう記号の出力確率は省略している．なお，開始状態からは直ちに次の状態に遷移する．また開始状態と最終状態では，記号の出力はない．

単語辞書には，各単語の「読み」，すなわち音素の系列が定義されている．

言語モデルとしては種々のモデルが提案されているが，本論文のシステムでは，FSA言語モデルを用いている．これは，単語の接続関係をFig. 2のように有限オートマトン(FSA)として定義したものである．ちなみに，Fig. 2の言語モデルでは，以下のような４つの文を受け付けることができる．

「頭ーがー痛い」
「頭ーがー悪い」
「おなかーがー痛い」
「おなかーがー悪い」

HViteではシステムが起動されると，その初期設定において，言語モデル，単語辞書，HMMから，Fig. 3のようなネットワークがメモリ上に作成される．

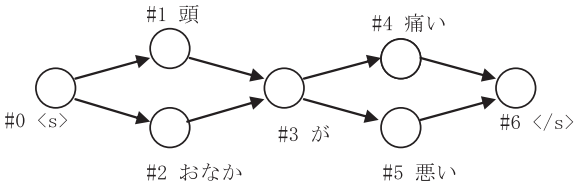

Fig. 2　FSA 言語モデル

単語をノードとし，それらの接続関係をリンクとしたグラフ構造で構成されている．#は各ノードの番号である．<s>, </s>はそれぞれ，開始，終了のノードであり，音声的には無音に対応する．

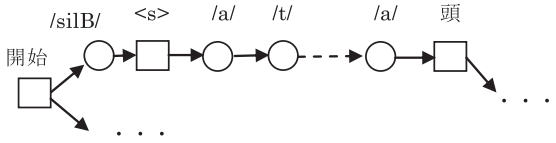

Fig. 3　認識用ネットワーク

丸は HMM ノード，四角は単語ノードである．/silB/は先頭の無音に対する HMM である．

1)　離散HMMの場合である．連続HMMの場合は，多次元の音響特徴量について，その確率分布（一般的には混合正規分布）が定義されている．

図において，丸はHMMノード，四角は単語ノードである．各HMMノードはさらにFig. 1のように構成されている．認識処理が開始されると，まず入力音声の最初の無音部分が/silB/[2)]のHMMを用いて認識される．認識ではHMMに定義された状態遷移確率に従って状態間の遷移が行われるが，同時に記号の出力確率を用いて入力音声とのマッチングが行われる．HMMの最終状態に到着すると直ちに後接する単語ノード（この場合は<s>）に遷移し（この時点で<s>が認識されたことになる），さらには<s>に続くHMMノード（この場合は「頭」「おなか」のそれぞれの先頭音素である/a/と/o/）の開始状態に遷移する．なお以下では，「現在，どの状態の処理を行っているか」をHTKでの言い方にならい，「トークンがどの状態に到着しているか」と表現することにする．処理は以降同様に行われ，例えばトークンが単語ノード「頭」に到着すれば，単語「頭」が認識されたことになる．入力音声とHMMとのマッチングにおいては，状態遷移確率，記号の出力確率からスコアが計算され，各トークンに格納される．なお，スコアとしては通常，確率の対数値（従ってマイナス値）が用いられる．また単語ノードに言語的なスコア（例えば，単語間の接続確率の対数値）が定義されていれば，単語が認識された時点でスコアに加算される．

ここで注意して欲しいのは，言語モデルでFig. 2のように複数のパスが定義されていれば，それらすべてのパスについて同時並行的に認識処理が実行されるということである．このため，処理すべきパスの数は増大し，現実的に処理することが困難になってしまう．この問題に対処するため，多くの音声認識システムでは，スコアが良い上位の候補だけを残し，それ以外の候補については以降の処理を中止（すなわち枝刈り）するようにしている．

上記の処理を発話の終わりまで行い，最終的に得られた候補のうち，最もスコアの良いものを認識結果として出力する．

以上の処理は，我々がベースシステムとして用いたHViteでもほぼ同じように実現されている．

３．漸次音声認識の処理方式

ここでは，我々が実現した漸次音声認識の処理方式について述べる．

３．１　言語モデルの木構造への展開

FSA言語モデルは，ある単語ノードで合流することを許したモデルである．このため，例えば，Fig. 2の例では，#3の「が」は，#1「頭」を経由してくる場合と#2「おなか」を経由してくる場合がある．このように複数のパスを経由して単語に到着する場合には，パスごとにスコアや到着時刻（フレーム番号）が異なるのが一般的である．そこで文献[1]と同様に，FSA言語モデルから木構造への展開を行い，パスごとの情報（認識されたフレーム番号，パスのスコア）を格納しておくようにする．なおこの展開は，認識時に動的に行うことにより，不要な展開をしないようにしている．

３．２　１段目のパスにおけるスコア

処理の進行に伴って認識パスが指数的に増大する理由は，言語モデルでのパスの組み合わせが増大することに加え，連続音声認識では可能性のある単語境界すべてについて認識を試みる必要があるためである．すなわち，ある１つの単語について終了フレームが異なる多数の結果が得られ，さらに次の単語は前単語の終了時点を開始時点として認識が開始されるため，パスの長さが長くなるに伴い認識パス数は膨大になってしまう．そこでこれらの第１段目のパスにおいて，FSA上の単語列上で同一のパスのものを時系列的にまとめ，２段目のパス情報として管理する．またパスごとの最良スコアを求めるために，１段目のパスのスコアの変化からその極大値（複数）を求め，その時点で各単語が認識されたとして最良スコアとフレーム番号を求める．なおこのため，出力するのはスコアが最大になった時点より若干遅れることになる．

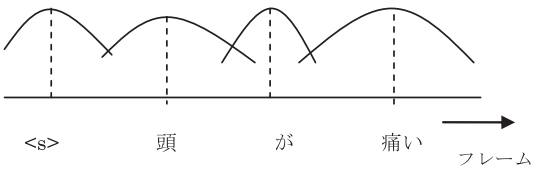

Fig. 4 部分パスのスコア
部分パスを構成する単語がフレームごとに認識される．単語のスコアはフレームの進行とともに変化する．

さらに，パスのスコアは先頭のフレームから単純に累積した値を用いるのではなく，累積スコアをフレーム数で割ったフレームあたりのスコアを用いる．２段目では，そのスコアのフレーム変化を見て最大になった時点でその単語が認識されたとして最大スコア値とともに出力するようにする．

実際のスコアのフレーム変化を観測すると，Fig. 4のように滑らかではなく，短時間での細かな増減がある．このため，まずスコアの平滑化を行う．以下のような移動平均により，平滑化スコアを求める．

2) /silB/は先頭の無音に対するHMMである．

$$sm[t] = \frac{1}{M}\sum_{i=0}^{M-1} s[t-i]$$

t : 時間 (フレーム番号)

$sm[t]$: 時間 t における平滑化後のスコア

$s[t]$: 時間 t におけるスコア

３．３　継続時間のスコアへの取り込み

我々が用いたHMM[5]では，各音素の継続時間は陽には定義されていない．そのため，このHMMをそのまま用いてもスコアの時間的変化がかなり小さく，最大値をうまく検出できないことが多い．そこでnモーラの単語の継続時間の分布は正規分布 $N(n\mu, n\sigma^2)$ であると仮定し，この確率から求めたスコアを全体のスコアに加える．なお，μ , σ^2 はあらかじめ学習用の音声コーパスより求めておく．

３．４　パスの枝刈りとスコアの最大値の検出

前述したように，時間の経過とともに１段目，２段目のパスともその数（特に１段目のパス数）は極めて多くなる．このため，スコアによりパスの枝刈りを行う．あるフレームにおいて，１段目のスコアが特定の閾値を下回ったものは破棄すると同時に，スコアの上位Nベスト候補を記録しておく．次に，各Nベストに対応する２段目のパスについて，スコアの最大値の検出を行い，最大値が検出されれば，出力を行う．具体的には以下のように処理する(Fig. 5).

（１）システムに1個，現フレームから過去Δフレーム（以降，これをフレーム幅と呼ぶ）のそれぞれに対して，Nベスト分の２段目パス情報へのポインタを格納する領域（すなわち，2次元配列）を用意しておく．

（２）新たに１段目のパスが生成（すなわち単語が認識）されたら，対応する２段目パスに１段目のパス情報（フレーム番号とスコア）を格納する．

（３）あるフレームにおいて（２）の処理が終わったら，スコアの良い上位N個（Nベスト）について，その２段目パス情報へのポインタを（１）で用意した２次元配列に登録する．

（４）（現フレーム－Δ）のNベストのそれぞれについて，（現フレーム－Δ）のb点を中心に，－Δの点(a)，＋Δの点(c＝現フレーム)を求め，各時点のスコアを比較して最良時点xとスコア$s(x)$のペア$(x, s(x))$を決定する（Fig. 5）．

（ケースA）$s(a) > s(b) > s(c)$の場合，
$(a, s(a))$とする．

（ケースB）$s(a) < s(b) > s(c)$の場合，
$(a, s(a)), (b, s(b)), (c, s(c))$の3点を通る2次曲線を考え，その最大値点$(m, s(m))$とする．

（ケースC）$s(a) < s(b) < s(c)$の場合，
まだスコアが良くなる可能性があるため，最大値点の取り出しは行わない．

（５）最良時点が決定された場合，[パス番号，単語名，フレーム番号，スコア]を出力する．また，そのパスについて，まだ未出力であれば 'N' のマークを付け，すでに1回以上出力されている場合は 'U' のマークを付けて出力する．

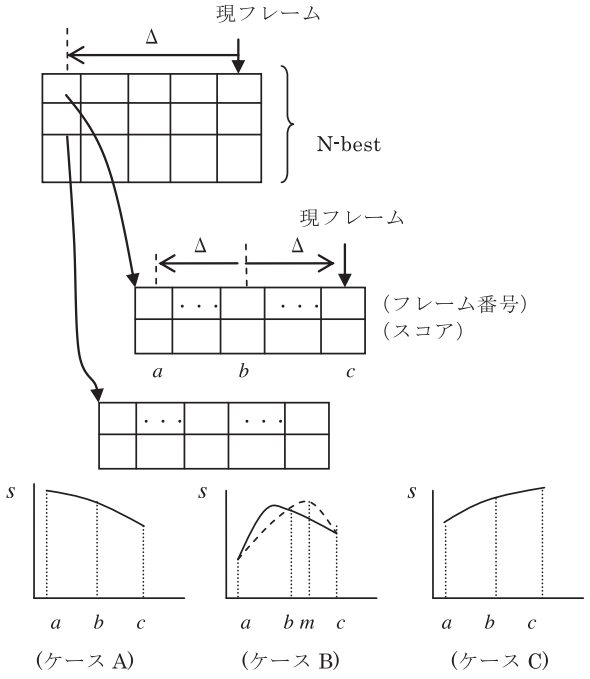

Fig. 5　N ベスト候補の抽出と最大値の検出

現フレームから，過去Δフレームの N ベストのパスを記憶しておく．パスでは，（現フレーム－Δ）の b 点を中心に，－Δの点(a)，＋Δの点(c＝現フレーム)を求める．各点のスコアを比較し，ケース A では a 点，ケース B では m 点を最大ポイントとする．ケース C では，最大ポイントは取り出さない．

３．５　出力インタフェース

Fig. 6のようなフォーマットで出力する．「前接のパス番号」が付加されているので，どのパスに接続する単語かを判断することができる．

N/U，パス番号，前接パス番号，順位，最大ポイントのフレーム番号，単語表記，スコア

Fig.6　　出力インタフェース

各項目の意味は以下の通り．

N/U：新規であれば N, 更新であれば U
パス番号：一意に付与された番号
前接パス番号 : このパスがつながる前のパスの番号
順位：N ベストにおける順位
最大ポイントのフレーム番号 : 最大スコアとなったフレームの番号
単語表記：認識された単語
スコア：認識された単語までのパスのスコア

４．動作例と考察

本提案方式の基本的な動作を確認する実験を行った．

１章でも述べたように我々の最終的な目標は，後段に適切な言語処理機能を配置し，発話の進行に伴って漸次的に内容を理解し，適当なタイミングであいづちなど返すことができるようなシステムを開発することである．従って，将来的には発話も漸次的に行われたもの（「．．．はですね，．．．」「．．．，えーと，．．．」などのような発話）を対象とすることを考えている．しかしまずは本システムの動作を確認するため，このような漸次的ではない（すなわち，途中での淀み等がない）発話，具体的には旅行ガイドブックに表れるような旅行会話文を対象とし，認識実験をおこなった．ガイドブック等から収集した300文から[6]の方法によりFSA言語モデルを作成し，また各文の音声発話を収録した．実験における諸元をTable 1に示しているが，N ベスト数は３，フレーム幅は５とした．さらに短い単語の挿入誤りに対処するため，1単語あたり一定値のペナルティを加えることとした．なお実際のペナルティ値は実験的に決定した．

実験全体ではかなりうまく動作したものと，あまりうまく動作しなかったものがあった．以下，それぞれの例について報告する．

（１）かなりうまく動作した例

かなりうまく動作した例をFig. 7しめす．入力された音声は「地下鉄-の-路線図-を-もらえ-ます-か」という発話である．なお，ここでは見やすさのため，3.5節で述べたフォーマットではなく，認識された単語列全体をその都度出力している．認識途中では間違った単語列も認識されているが，発話が進むにつれ，正しい単語列に徐々に収束していくことが分かる．正解の先頭単語「地下鉄」は88フレームあたりで絞り込まれて（確定して）いる．また「地下鉄-の-路線図」までは133フレームあたりで確定している．これから，本方式ではかなり早い段階で正しい候補を出力できていることが分かる．また Fig. 7で出力された正解単語（<s>, </s>を除く）についての最良時点（フレーム番号）と，発話完了後に最尤確定された単語の認識時点との誤差の絶対値は，単語あたりの平均で3.15フレームであり，フレーム間隔が10ミリ秒であるから，約32ミリ秒の誤差であった．また確定するまでには，さらに5フレームの遅れが加わるから，誤差と合わせて最大82ミリ秒遅れることになる．

Table 1 音声認識実験の諸元

HMM	4 ミクスチャのトライフォンモデル
言語モデル	旅行会話文 300 文から作成した FSA 言語モデル 単語ノード数=878 平均ブランチ数=1.53
テスト発話	上記会話文の音声発話を収録（男性 3 名が各々別文を発話）
認識条件	平滑化のための移動平均数=10 N ベスト数=3 フレーム幅=5

（２）あまりうまく動作しなかった例

あまりうまく動作しなかった例での入力音声は「美術館-めぐり-の-ツアー-は-あり-ませ-ん-か」という発話である．途中フレームで「美術館-めぐり-の」までは確定したが，それより後ろの部分は発話全体の認識が終わるまで確定しなかった．これは文末に近づくにつれ，長さの短い単語が続き，また発話の明瞭性等が低下したのが原因であろうと思われる．ただし発話終了後の最尤確定処理では正しい認識結果が得られた．

５．まとめ

入力発話の終了を待たず，認識した結果を漸次的に部分木として出力する音声認識システムの処理方式について報告した．また実際に認識実験を行った結果について報告した．今後は発話自体も漸次的に行われたものについて認識実験を行う予定である．また，漸次理解を行うことのできる言語処理部についても開発を進めていく．

【参考文献】

[1] P. F. Brown, J. C. Spohrer, P. H. Hochschild and J. K. Baker: “Partial Traceback and Dynamic Programming,” Proc. of ICASSP-82, pp.1692-1632, 1982

[2] 今井，田中，安藤，磯野：「最ゆう単語列逐次比較による音声認識結果の早期確定」，電子情報通信学会誌，D-II, Vol.J84-D-II, No.9, pp.1942-1959, 2001

[3] 中川，小林：「連続単語認識における部分単語列の早期検出」，音講論集，3-1-8，pp.97-98，1998

[4] S. Young, et al.: “The HTK Book (for Ver. 3.0),” 1999 (http://htk.eng.cam.ac.uk/　2013.4現在)

[5] T. Kawahara, A. Lee, K.　Takeda, K. Itou, and K. Shikano: “Recent Progress of Open-Source LVCSR Engine Julius and Japanese Model Repository -- Software of Continuous Speech Recognition Consortium,” Proc. of ICSLP-2004, 2004
(http://julius.sourceforge.jp/en/julius.html　2013.4現在)

[6] T. Morimoto, S. Takahashi: "Automatic Construction of FSA Language Model for Speech Recognition by FSA DP-Matching,” Lec. Notes in Electrical Engineering, Vol.6, Springer, 2008

```
File: ../Data/Raw/12.raw
  N: Frame=22 rank=1 frame=18: <s>-: -58.33
  N: Frame=31 rank=3 frame=26: <s>-１-: -65.66
  N: Frame=50 rank=3 frame=45: <s>-スカート-: -62.86
  N: Frame=51 rank=2 frame=47: <s>-貸し-: -62.86
  N: Frame=61 rank=1 frame=57: <s>-地下鉄-: -60.97
  N: Frame=72 rank=2 frame=67: <s>-地下鉄-の-: -61.15
  N: Frame=75 rank=3 frame=65: <s>-ランプ-: -65.97
  N: Frame=88 rank=2 frame=83: <s>-地下鉄-の-ある-: -62.83
  N: Frame=100 rank=2 frame=97: <s>-地下鉄-の-バス-: -62.79
  N: Frame=109 rank=3 frame=106: <s>-地下鉄-の-保険-: -62.13
  N: Frame=121 rank=1 frame=117: <s>-地下鉄-の-路線図-: -60.23
  N: Frame=121 rank=2 frame=117: <s>-地下鉄-の-路線図-: -60.23
  N: Frame=129 rank=3 frame=119: <s>-地下鉄-の-話せる-: -62.78
  N: Frame=131 rank=3 frame=121: <s>-地下鉄-の-シングル-: -62.58
  N: Frame=133 rank=1 frame=123: <s>-地下鉄-の-路線図-を-: -60.45
  N: Frame=139 rank=2 frame=129: <s>-地下鉄-の-路線図-は-: -61.05
  N: Frame=153 rank=3 frame=146: <s>-地下鉄-の-路線図-を-２-: -61.92
  N: Frame=154 rank=2 frame=151: <s>-地下鉄-の-路線図-は-買え-: -61.63
  N: Frame=155 rank=1 frame=151: <s>-地下鉄-の-路線図-を-もらえ-: -60.59
  N: Frame=178 rank=1 frame=174: <s>-地下鉄-の-路線図-を-もらえ-ます-: -60.95
  N: Frame=178 rank=2 frame=168: <s>-地下鉄-の-路線図-は-閉まっ-: -61.93
  N: Frame=178 rank=3 frame=168: <s>-地下鉄-の-路線図-を-お願い-: -61.97
  N: Frame=195 rank=1 frame=192: <s>-地下鉄-の-路線図-を-もらえ-ます-か-: -60.92
Recognition result:
<s> 地下鉄 の 路線図 を もらえ ます か </s>  ==  [197 frames] -61.0463  [Ac=-11326.1
LM=-700.0] (Act=4190.1)
```

① ②

Fig. 7　認識例（かなりうまく動作した例）

①の部分が逐次出力された部分
　左より，以下の事項をしめす.
　　[N(ew)/U(pdata)の別，処理時のフレーム番号，順位，最大値ポイントのフレーム番号，単語列，スコア]
②は，発話完了後に最尤確定された認識結果