TR-I-0085

TDNN音韻スポッティングと拡張LRパーザを用いた文節音声認識

Continuous Speech Recognition Using TDNN Phoneme Spotting and Generalized LR Parser

南泰浩 宮武正典 沢井秀文 鹿野清宏
Y.MINAMI M.MIYATAKE H.SAWAI K.SHIKANO

1989.7.4.

概要

本報告ではTDNN(時間遅れニューラルネットワーク)音韻スポッティングと拡張LRパーザを用いた文節認識システムを構築し、その基本的な構成について種々の実験を行い、検討した結果について述べる。本システムは、拡張LRパーザを音韻予測に使い、その予測された音韻をTDNNスポッティング結果とDPマッチングにより評価し、認識結果を出力するものである。ここでは、TDNNの学習データ、DPの距離尺度、継続時間制御などを変化させた場合に文節認識率がどのように変わるかについて述べる。本システムで各種のパラメータを改良した結果、短い文節(SB3)での文節認識率63.4%を得た。

ATR 自動翻訳電話研究所
ATR Interpreting Telephony Research Laboratories

目　次

１．　はじめに

近年、神経回路の構造を模倣したニューラルネットワークの研究が盛んに行われるようになってきた。これらの研究の中で特にＲｕｍｅｌｈａｒｔのＢａｃｋ－Ｐｒｏｐａｇａｔｉｏｎアルゴリズムはこれまで困難であった多層型のパーセプトロンの学習を可能にした。このアルゴリズムは最近では各種のアプリケーションに適応され、様々な研究成果が発表されている。

現在、音声認識の分野においてもＢａｃｋ－Ｐｒｏｐａｇａｔｉｏｎを利用した様々な研究がなされている。この中でＴＤＮＮ（時間遅れニューラルネットワーク）は、音韻の特徴を抽出する構造になっている、入力の位置ずれに強いなどの特徴を有し、連続音声の認識に有利である[1][2]。

このニューラルネットを連続音声に適応する場合には言語情報を使うことが必要となる。現在非常によく用いられる文法は文脈自由文法と呼ばれる文法で、この文法の解析手法は、いくつか考案されている。この中でＬＲパーザは構文解析を決定的に実行できるため、高速な処理が可能である。しかし、ＬＲパーザは文脈自由文法の一部の文法しか解析できず、曖昧な文法を処理することができない。拡張ＬＲパーザでは構文を並列に解析することでこの問題点を解決した。

本報告ではＴＤＮＮと拡張ＬＲパーザを利用した連続音声認識システムについて述べる。本システムは拡張ＬＲパーザを音韻予測に使い、その予測された音韻とＴＤＮＮスッポティング結果からＤＰマッチングにより評価し、認識結果を出力する。

２．　ＴＤＮＮによる音韻スポッティング[1][2]

ＡＴＲでは、音声現象を考慮した図２－１に示すようなＴＤＮＮを考案した。このシステムは／ｂｄｇ／の音韻認識において高い認識率を達成した。このＴＤＮＮは学習としてBack-Propagationアルゴリズムを用いている。

このＴＤＮＮには以下の３つの特徴がある。

（１）４層のNeural Networkの構成であるので、任意の判別境界をBack-Propagationアルゴリズムにより構成することが原理的に可能である。

（２）ＴＤＮＮは、学習により音韻のスペクトログラムから、音韻特徴を発見して、音韻を認識することを意図した構成となっている。

（３）ＴＤＮＮは、入力音韻の位置ずれに影響されにくい構成となっている。

２．１．　全音韻に対するＴＤＮＮ

本報告で用いるＴＤＮＮは以上で述べたＴＤＮＮを拡張しすべての音韻（２４カテゴリ）に拡張した形となっている。この構成を図２－２に示す。このネットワークはかなり規模の大きな構成となるため、通常のBack-Propagationアルゴリズムでは非常に多くの計算量を必要とする。そこで、本報告ではこのＴＤＮＮの

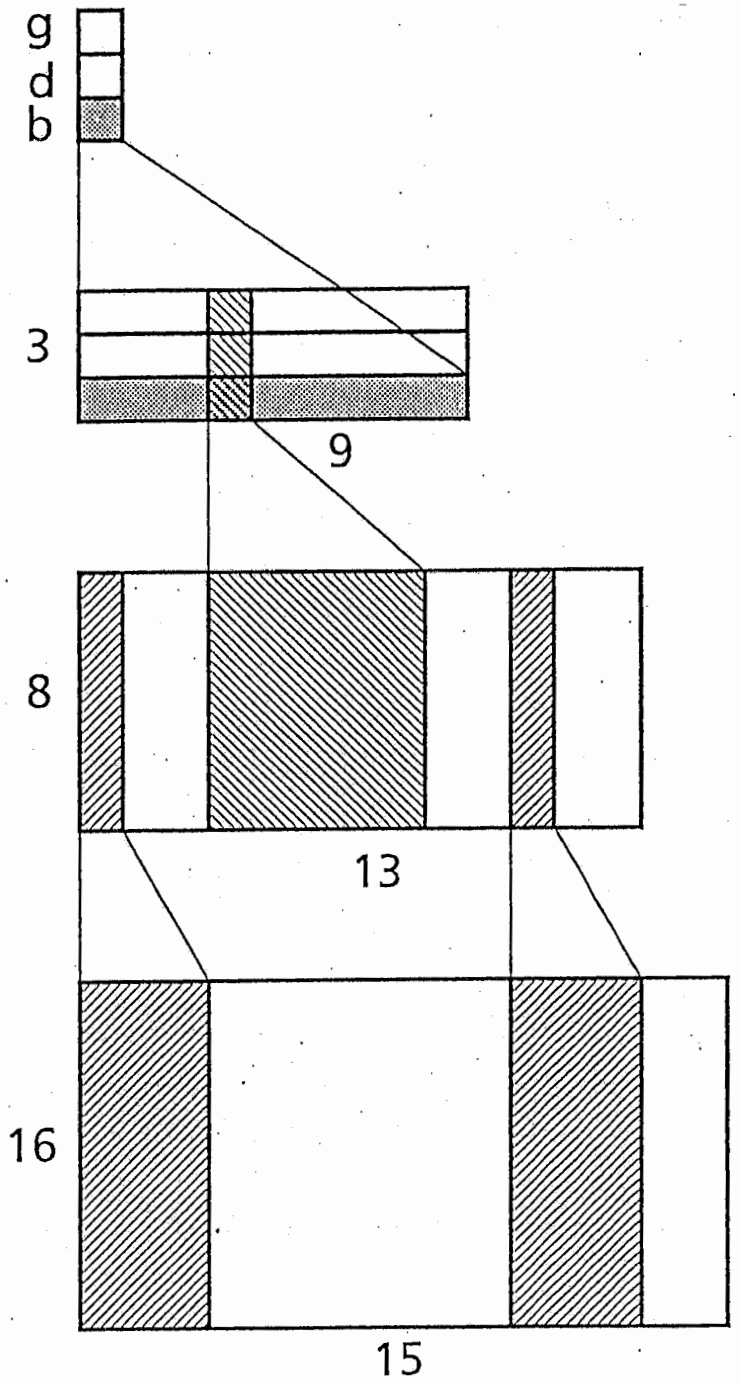


図2-1 TDNN

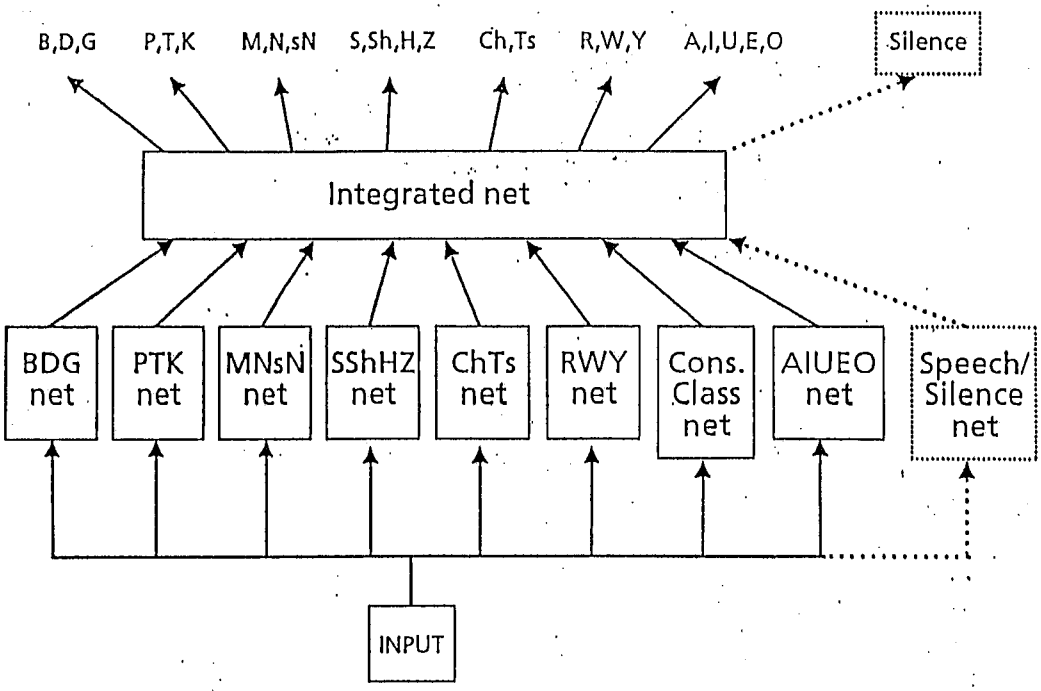


図2-2　24音韻のTDNN

構成を行うために高速なアルゴリズムを採用する[4]。本報告ではさらに従来の教師信号以外の教師信号を用いた学習についても検討を行う。

２．２．　音韻スポッティング

２．１で述べたＴＤＮＮを用いて文節中の音韻のスポッティングを行う。この手法の構成図を図２－３に示す。図中下側はＴＤＮＮの入力層と入力音声を、上側は出力層と出力結果を示す。いずれもＴＤＮＮの入力層は１６次＊１５フレームの構成で、１５０ｍｓｅｃ分の音声を受け付けるようになる。このような操作を入力音声に対して１フレームずつシフトし、その出力値をスッポティング結果とした。本報告ではこの２４音韻のＴＤＮＮの結果を用いる前に、対称とする区間が音声であるか無音であるかを判別するネットワークを使用する。無音であれば２４音韻のＴＤＮＮの出力を用いないようにしている。このスポッティング方法の概略を図２－４に示す。スポッティング結果の例を図２－５に示す。

２．３．　ＴＤＮＮの教師信号

現在Back-Propagationの学習としては目的値とネットの出力値の差の自乗距離を用いている。しかし、この関数は識別平面の境界付近のデータに対しても強制的に出力を１または０にするように学習する。このため境界付近では曖昧な中間値をとることができない。そこでここではいくつかの下に示す関数を用いてこれを回避する手法を検討する。

（１）ＭＳＥ

定義　t_i：パターンｐの出力層ｉの目標値

o_i：ネットの出力層ｉの出力値

$$E = \frac{1}{2} \sum_i (t_i - o_i)^2$$

この関数をBack-Propagationで最小にする様に学習する。

（２）ＣＦＭ[5]

定義　o_t：１になってほしい出力層の出力

o_n：０になってほしい出力層の出力

Δ_n：$o_t - o_n$

$$CFM = \sum_n \frac{\alpha}{1 + \exp(-\beta \Delta_n + \zeta)}$$

この関数をBack-Propagationで最大にする様に学習する。

（３）弱者救済法

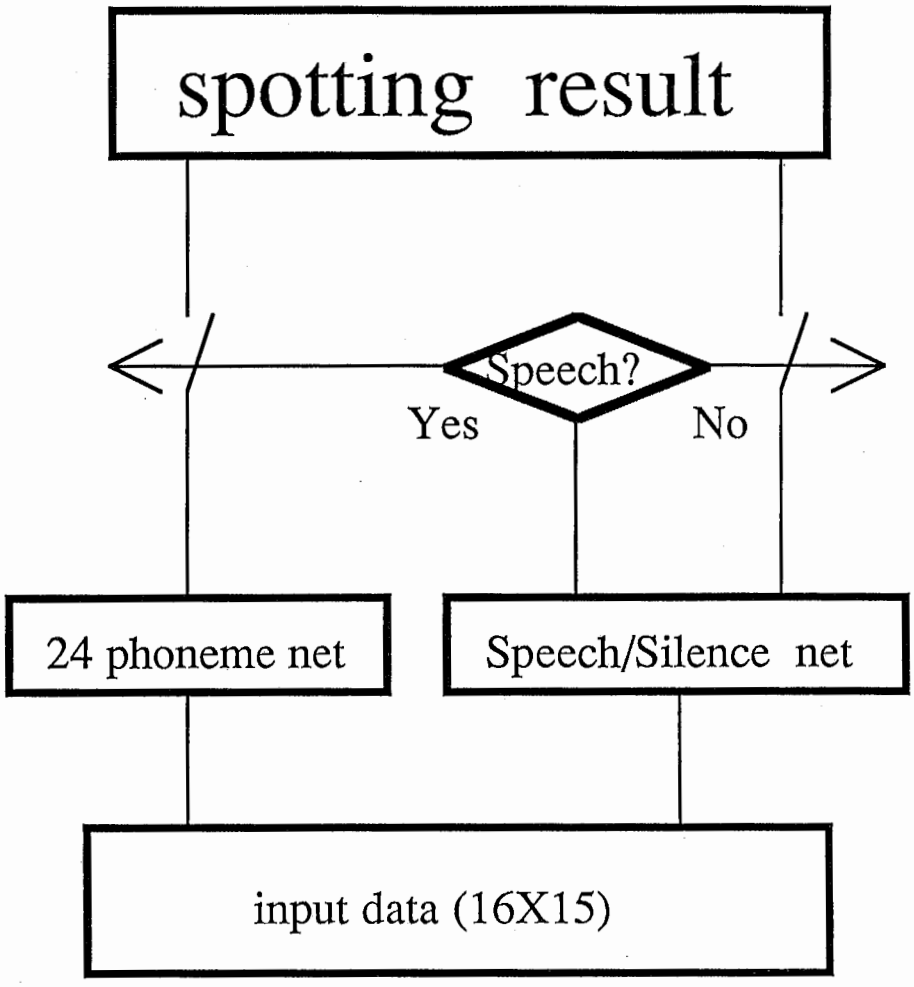


図2-3 スポッティング構成

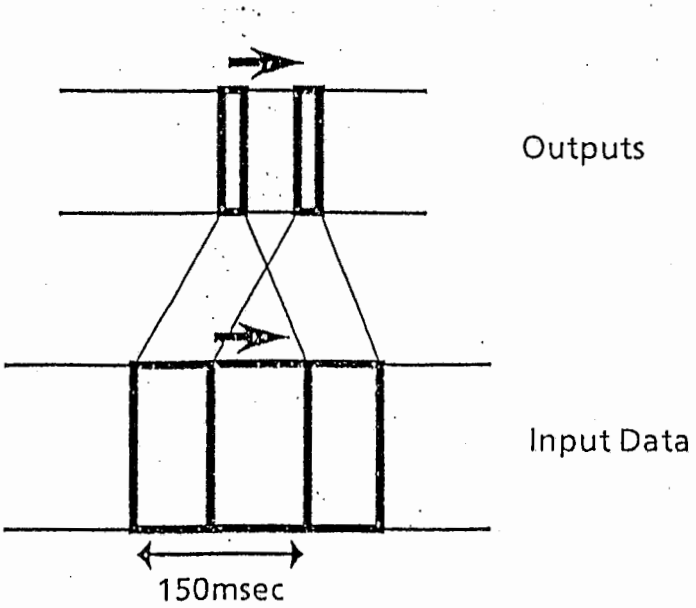


図2-4　スポッティング方法

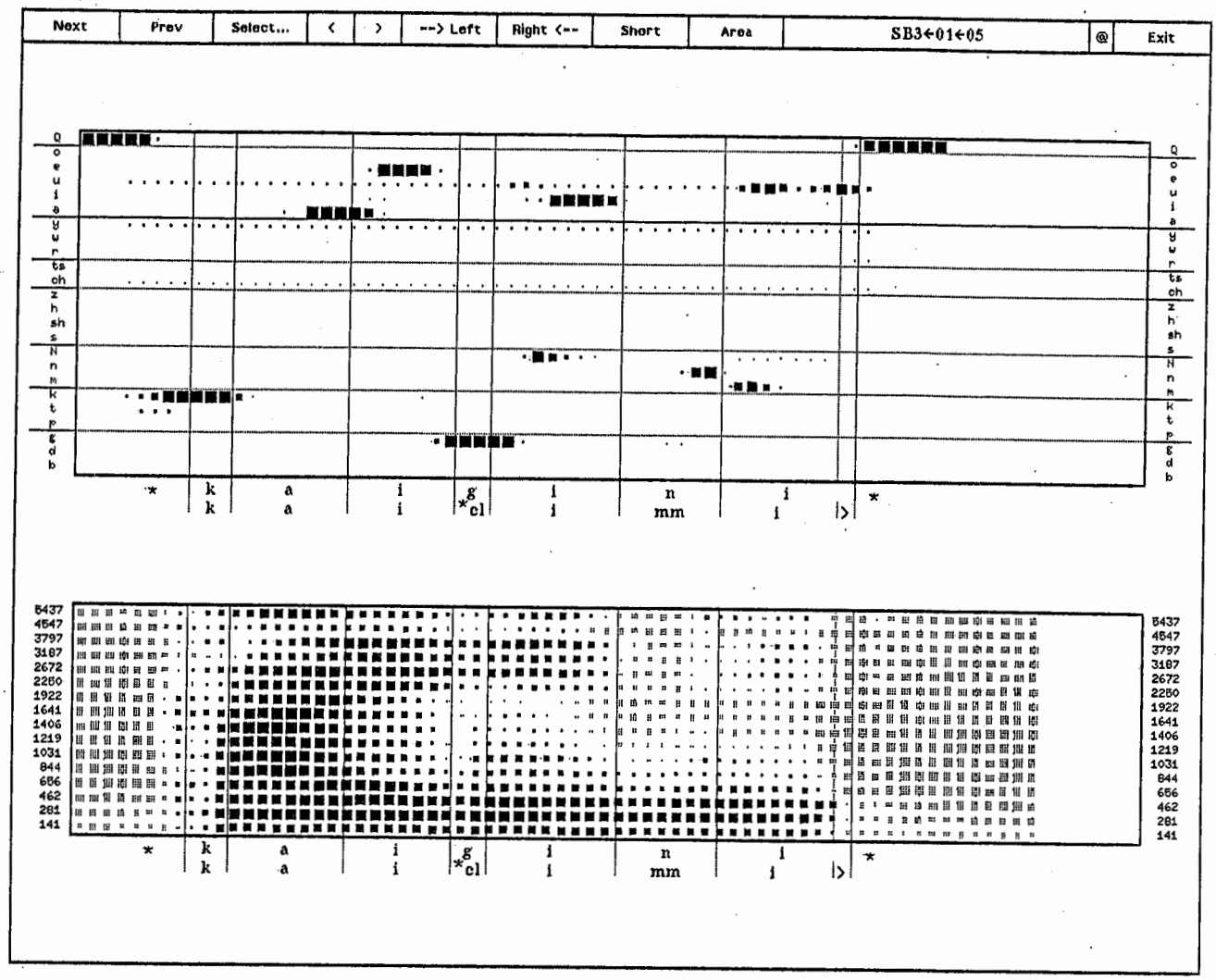


図2-5　スポッティング結果の例
(発声は"会議に")

FUNC = E if ($0.3*o_t < \max(o_n)$)

FUNC = 0 else

この関数を用いて学習を行う。

3. 拡張ＬＲパーザによる構文解析[3]

3．1. ＬＲパーザ

ＬＲパーザは文脈自由文法の中で、ＬＲ文法という限定された文法から生成される文法を解析できる。このパーザは入力記号を受付けながらバックトラックなしに決定的に構文を解析できる。

ＬＲパーザは動作表と行先表という２種類の表を見ながら解析を行う。動作表は次にパーザが行う動作を示す表であり、行先表は次にパーザがとる状態を示す表である。

パーザの動作には次に挙げる４種類がある。

（１）移動（shift）
（２）還元（reduce）
（３）受理（accept）
（４）誤り（error）

移動はパーザの状態をスタックに積む動作で、還元はスタック上の記号を文法規則に従ってまとめるものである。また受理は、入力文章がＬＲパーザで解析できたことを示し、誤りは解析できなかったことを示す。

次に、解析の手順を示す。

定義

s　　：パーザの状態
a　　：文法記号（非終端、終端記号）
入力ポインター：現在処理中の入力記号列を示す。
状態スタック　：パーザの状態を保存する。
ＧＯＴＯ（ｓ，ａ）：状態ｓと文法記号ａから次の状態を求める。
ＡＣＴＩＯＮ（ｓ，ａ）：状態ｓと文法記号ａからパーザの動作を求める。

＜アルゴリズム＞

（１）初期化

入力ポインターを入力記号列の先頭に位置づける。状態スタックに０をプッシュする。

（２）現在の状態ｓと入力ポインターの示す記号ａからＡＣＴＩＯＮ（ｓ，ａ）を調べる。
（３）ＡＣＴＩＯＮ（ｓ，ａ）＝”ｓｈｉｆｔ”ならばＧＯＴＯ（ｓ，ａ）を状態スタックにプッシュし、入力ポインターを一つ進める。
（４）ＡＣＴＩＯＮ（ｓ，ａ）＝”ｒｅｄｕｃｅ，ｎ”ならば、ｎ番目の文法規則の右辺にある文法記号の数だけスタックの状態をポップする。スタック最上段の状態をｓ’とすると、ｓ’とｎ番目の文法規則左辺にある文法規則Ａから、次の状態ＧＯＴＯ（ｓ’，Ａ）を求めスタックにプッシュする。
（５）ＡＣＴＩＯＮ（ｓ，ａ）＝”ａｃｃｅｐｔ”ならば解析終了。
（６）ＡＣＴＩＯＮ（ｓ，ａ）＝”ｅｒｒｏｒ”ならば解析失敗。
（７）（２）に戻る。

３．２．　拡張ＬＲパーザ

拡張ＬＲパーザはＬＲパーザでは対処できなかった曖昧な構文を解析できるようにしたものである。拡張ＬＲパーザでは動作表に複数の項目を記述する。パーザがこの複数の項目の表を調べた場合には並列動作を行う。このようにして決定的に構文の解析を行う。

４．　ＴＤＮＮ－ＬＲの構成

図４－１にＴＤＮＮ－ＬＲの基本構成を示す。文法規則はあらかじめ文脈自由文法より各表として登録する。ＬＲパーザでは３．で示したような解析を行うのではなく、いままで処理された音韻系列から次の音韻系列を予測する。この予測された音韻とＴＤＮＮのスポッティング結果に対しＤＰマッチングの手法を用いてその音韻の評価を行う。

４．１．　音韻予測

ここで用いる音韻予測ＬＲパーザはある状態ｓでの動作表を調べ受け付けることが可能な音韻をすべて求める。この音韻は文法で規制された制限下での予測された音韻となる。ここで用いるＬＲパーザの文法の終端記号は音韻となる。

４．２．　評価方法

予測された音韻をＴＤＮＮのスポッティング結果との間でＤＰマッチングを行う。ＤＰを行うためには標準パターン長と、標準パターンと入力パターン間の距離尺度を決定する必要がある。予測された音韻の平均の継続時間長をあらかじめ求めておき、この長さを標準パターン長とする。あとは予測された音韻とＴＤＮＮのスポッティング結果との距離を求めればよい。距離尺度については５．で述べる。

以下にこれらを求めるアルゴリズムを述べる。

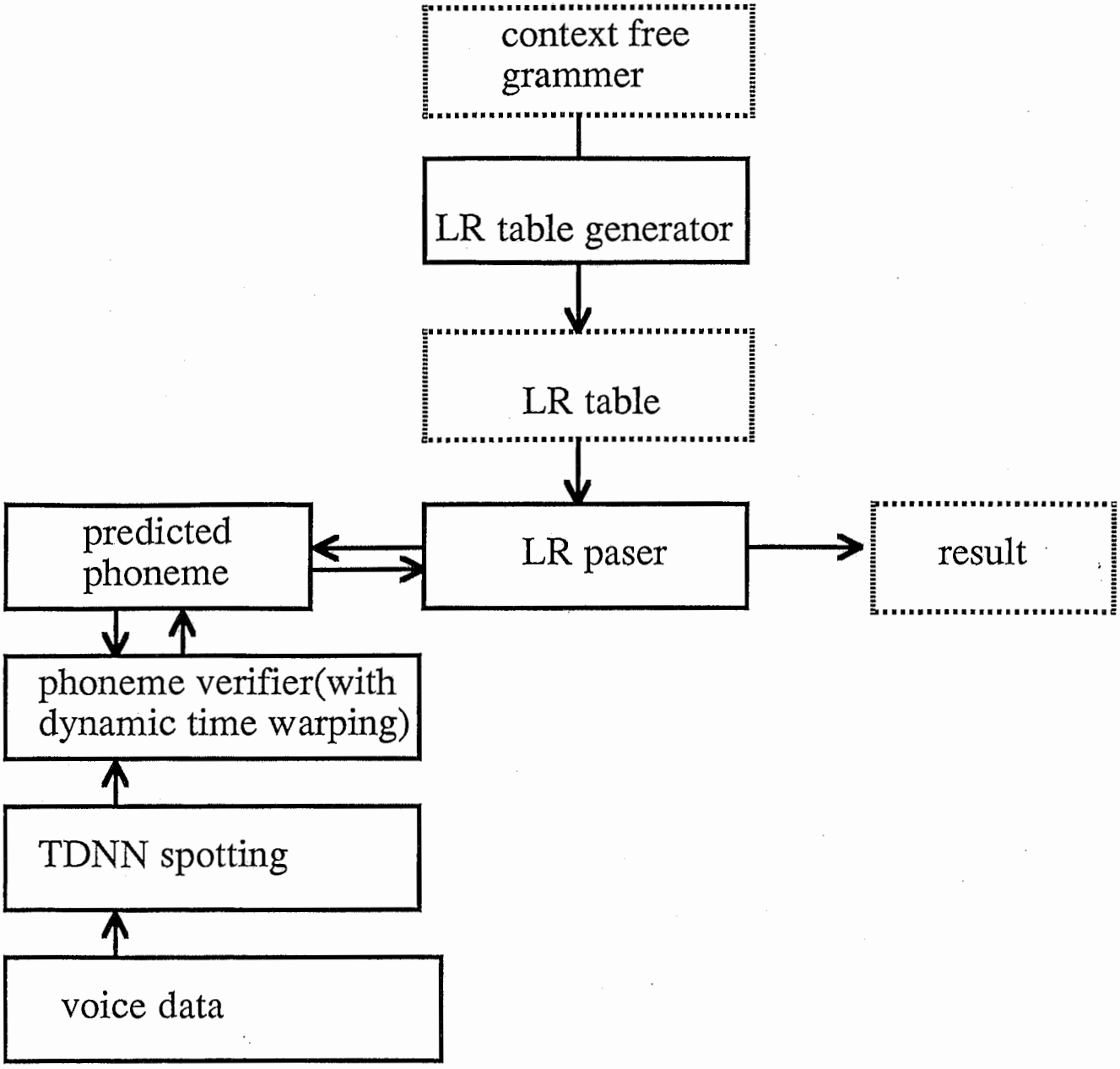


図4-1 TDNN-LRの構成

＜アルゴリズム＞

ここでは音韻照合結果を効率よく管理するためにセルというデータ構造を用いる。このセルには以下の情報が保存される

（１）ＬＲパーザの状態スタック

（２）ＤＰの累積距離保存用テーブル

（３）受理マーク

ＴＤＮＮ－ＬＲでは以下のような手順で認識を行う。

（１）初期化

新しいセルＰを１つ作り、ｐのＬＲ状態スタックに０をプッシュし、ｐの受理マークをｏｆｆにする。また、累積距離保存用テーブルも初期化を行う。

（２）セルの分岐

Ｓ＝｛（Ｃ，ａ，ｘ）｜Ｃis a cell & Ｃ is not accepted & s is a state of Ｃ & x(ＡＣＴＩＯＮ（ｓ，ａ）＝ｘ＆　ｘ！＝”ｅｒｒｏｒ”）｝

を求める。集合Ｓの各要素に対して以下を行う。集合Ｓが空ならば終了する。

（３）ｘ＝”ｓｈｉｆｔ”ならば音韻ａをＤＰにより照合を行う。

ここでのＤＰ照合はいくつかの手法を試みた。ここで各フレーム毎の累積距離を計算し、この値を累積距離保存用のテーブルに保存する。累積距離を入力フレーム数あるいは、参照パターン長で正規化した値の最大値を求める。この値がある閾値より大きい（小さい）ときにはこのＣを捨てる。そうでなければこのセルＣのＬＲ状態スタックに新しい状態をプッシュする。

（４）ｘ＝”ｒｅｄｕｃｅ　ｎ”ならばｎ番目の構文規則によって還元操作を行う。

（５）ｘ＝”ａｃｃｅｐｔ”であり正規化累積距離の最大値のフレームが入力の最後尾よりある値以内にあればセルＣの受理マークをｏｎにする。

ここでは、この手法による解空間が比較的大きくなるのを避けるためにビームサーチを行う。これはセルＣを正規化累積の順にソートし、小さい（大きい）ものからＢ個をとってくるようにすればよい。

5．　実験

5．1．　実験条件データ

1名の話者によって文節単位に発声された25文章279文節で、各文節の構造は、1個の自立語の後に複数（0個も含む）の付属語が連鎖したものとなっている。この音声データは、サンプリング周波数12KHzでAD変換後、フレーム周期5msごとにハミング窓をかけて、15次メルスケール化し、フレーム周期10msに統合したものである。

5．2．　実験に用いた項目

5．2．1．　距離尺度

（1）参照パターンを予測された音韻の要素が1で他の23音韻が0の24次元ベクトルと考える。この参照パターンとスポッティング結果の24次元ベクトルのユークリッド距離を距離尺度とする。
（2）スポッティング結果を確率と考え、予測された音韻のスポッティング結果の対数を距離尺度とする。この場合DPパスは最大値を求めることになる（対数をとるのは後に述べる単語の時間長制御との整合性を考えたためである）。

5．2．2．　DP傾斜制限

ここでは以下に述べる3種類の傾斜制限を用いた。これを図5－1に示す。
（1）参照パターン基準の非対称。
（2）入力パターン基準の非対称。
（3）入力パターン基準の非対称。

5．2．3．　時間長制御

時間長制御はDPの計算を全て行った後、音韻の継続時間長からガウス分布によって確率を計算し、DPの累積距離に掛け合わせる。
ガウス分布は5240単語の語頭、語中、語尾、全体の4種類を用いた。ガウス分布、ガウス分布の自乗の2つの時間長制御について実験を行った。

5．2．4．　TDNN学習用音声データ

（1）音韻境界のみのデータ4800個。
音韻境界部のデータは5－2図に示す用に5240単語のデータから切り

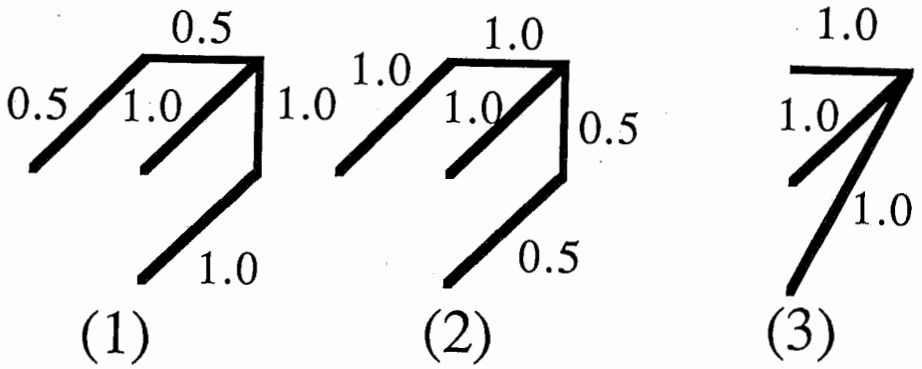


図5-1 DPパス

出しランダムに各音韻２００個づつ４８００個のデータを使用した。

（２）音韻境界（子音）と音韻中心（母音＋Ｎ）４８００個

上で用いた音韻のうちで母音とＮ（撥音）に対しては音韻の中心を切り出したもの　を用いた。切り出し位置は図５－３に示す。

（３）音韻全域データ９６００個

音韻全域の学習データで、イベント層の両側の境界部分１０ｍｓを除いたものを使用する。このデータを各音韻カテゴリ毎に、４００個用いる。

（４）音韻全域２４０００個

（３）で示した、データを各音韻カテゴリ毎に、１０００個用いる。

５．２．５．　ＴＤＮＮの出力の補正

ＴＤＮＮ（学習にＤｙｎｅｔを使用した場合）の出力値は図５－１のようにある値が定常的に加算された出力となっているこのためこの値を補正する必要があると考えられる。本報告ではこの必要性を確認するため、以下に示す手法を用いた。

（１）補正を行わない。

（２）音韻ごとにオフセット値を調べこの値を引いた値を出力とする。

（３）全音韻に対してある一定の値を引いたものを出力値とする。

５．２．６．　Ｓｐｅｅｃｈ／Ｓｉｌｅｎｃｅネットの存在について

（１）ネットをいれない。

（２）ネットをいれる。

５．２．７．　ｐｔｋの前のｃｌの処理

通常ｐｔｋの前にはｃｌ（ｃｌｏｓｕｒｅ）が存在することが多い。そこでこのｃｌの存在を考慮してｐｔｋの部分に対しては、ｃｌとのＤＰを行った後に、ｐｔｋのマッチングを行うようにした。これに関しては以下の２つの条件で実験を行った。

（１）ｃｌの処理を行わない。

（２）ｃｌの処理を行う。

５．３．　実験結果

実験は５．２で述べた項目のいくつかの組合せについて行った。

５．３．１．　音韻境界のみのデータに関する検討

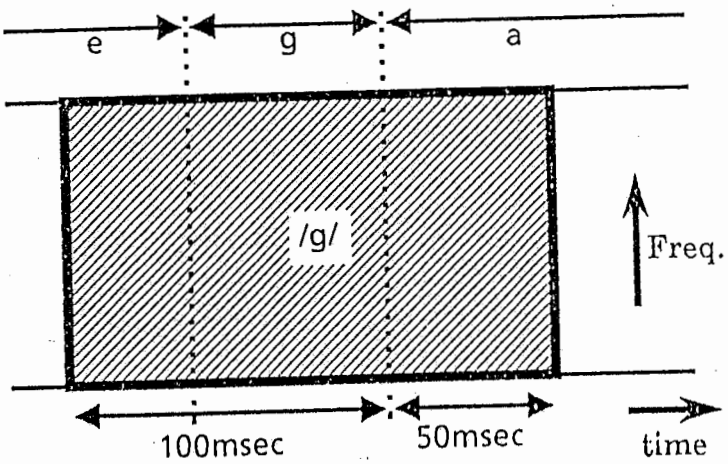


図5-2 音韻境界データ(/g/)

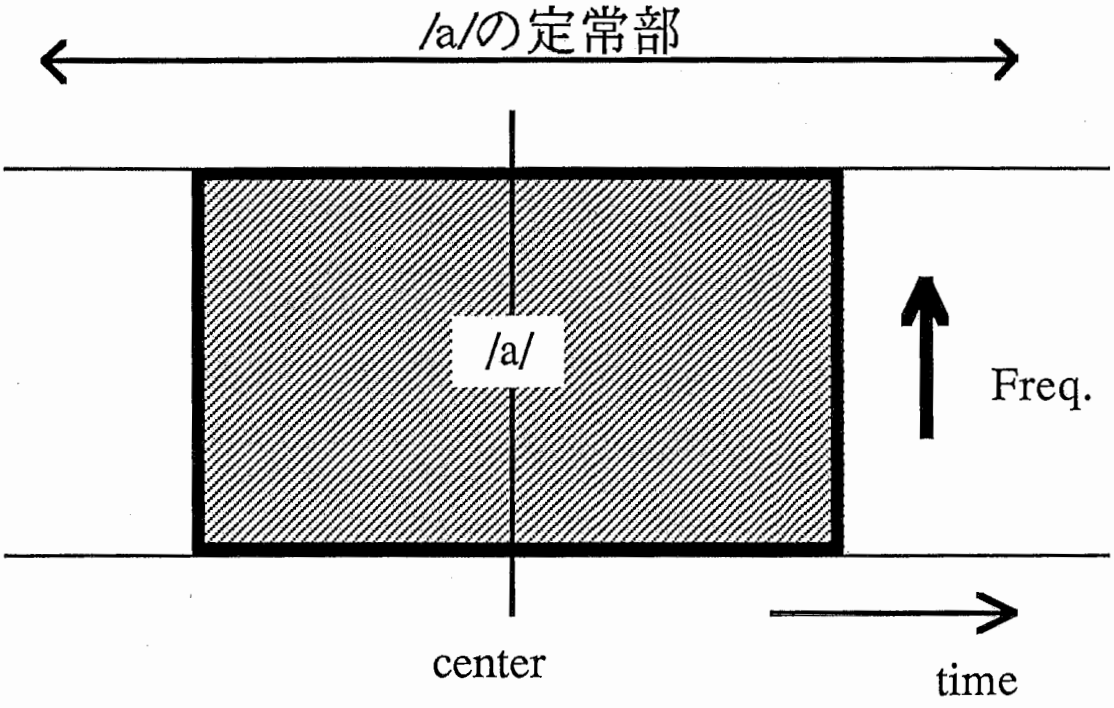


図5-3 音韻中心データ(/a/)

実験条件　（１）音韻境界のデータのみを学習に使用した場合。
（２）出力の補正一定値以下のデータは全て０とした。
（３）ビーム幅（Ｂ）はすべて１００。
（４）ｐｔｋの前のｃｌを考慮。
（５）Ｓｉｌｅｎｃｅ／Ｓｐｅｅｃｈネットあり。

以下の実験は上で述べた実験条件のもとに行った。

（１）距離尺度による検討

距離尺度を５．２．１．で述べた２種類の方法で試みた。結果を図５－４に示す。この図から対数を使った場合の方が１位の認識率ではよいことがわかる。ただし５位までの認識率では、ユークリッド距離を用いた方がよかった。これはｆｌｏｏｒｉｎｇの調整を対数の方で行っていないためと考えらる。

（２）時間長制御に関する検討

時間長制御についてはガウス、ガウスの自乗を用いたものを比較した。この結果を図５－５に示す。この結果から時間長制御を行う方がよいことが確認された。

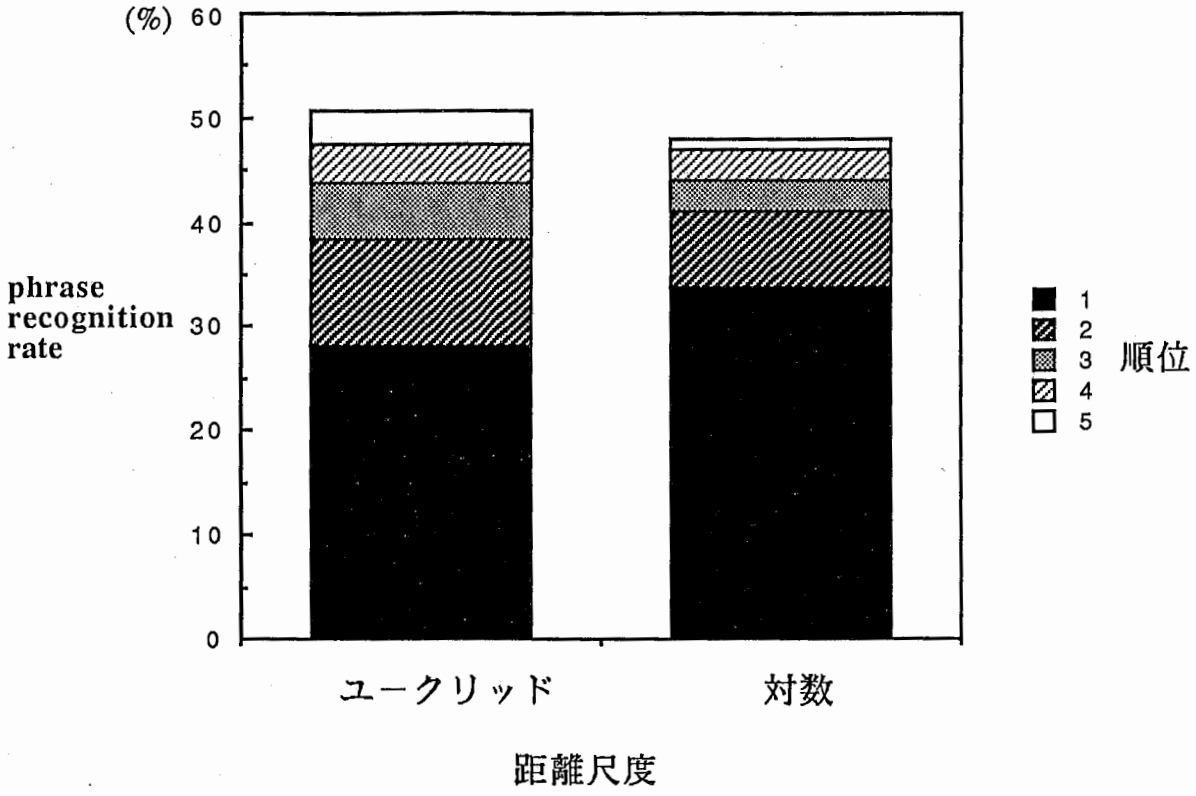


図5-4 距離尺度による文節認識率

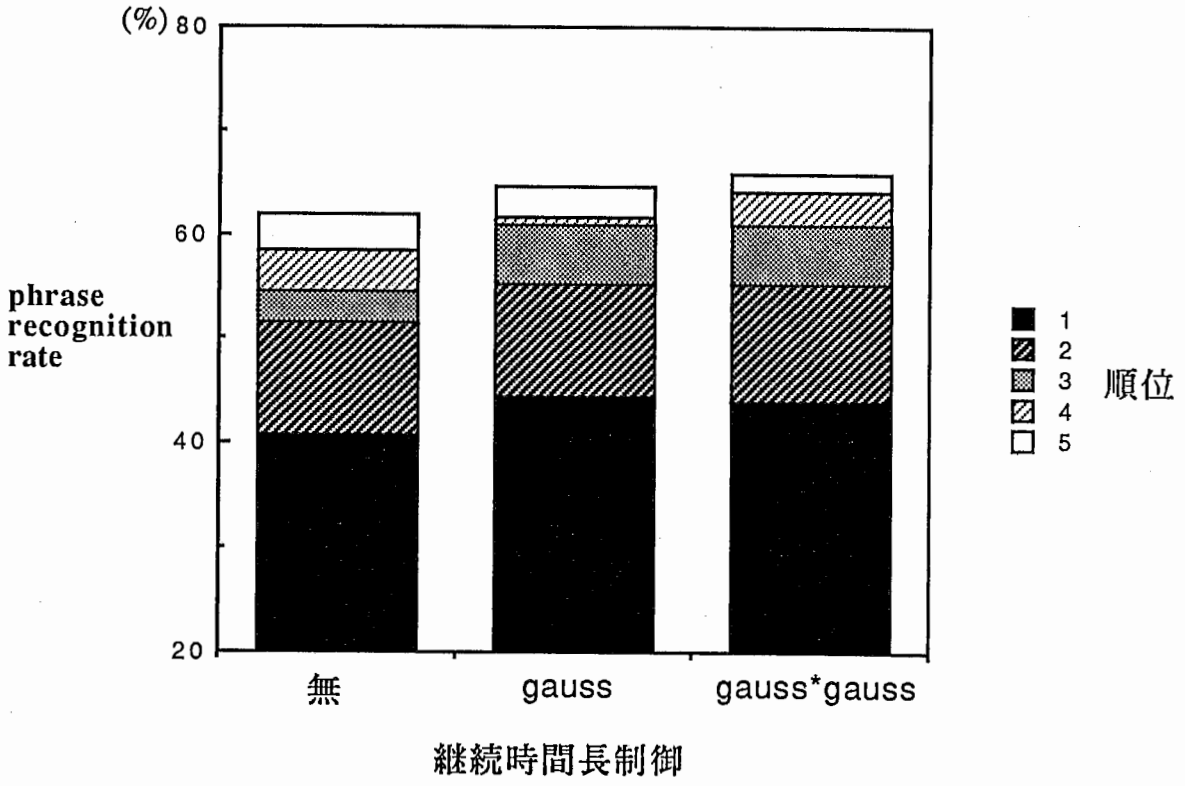


図5-5 時間長制御による文節認識率

（３）ＤＰパスによる検討（１）

５．２．２．で示した３種類の内２種類についての比較を行った。この結果を図５－６に示す。この実験で（１）のＤＰパスを使った方が認識率がよいことが確認された。

（４）目的関数の検討

学習方法をＣＦＭに変えた実験を行った。学習条件は$\alpha=1$，$\beta=4$，$\xi=1$である。この結果を図５－７に示す。この結果からＣＦＭとガウスの自乗の継続時間長制御を行ったものが最も認識率がよかった。ＣＦＭではＭＳＥで出力されなかった音韻が出力されるようになる（図５－８、図５－９参照）。このため挿入なども多くなるので、きつい時間長制御が有効であると考えられる。

なお、ＣＦＭを使ったものに関しては上の実験条件（２）出力の補正は行っていない。

弱者救済法についても実験を行ったが図５－１０に示すようなスポッティング結果となり、有効でないことがわかった。

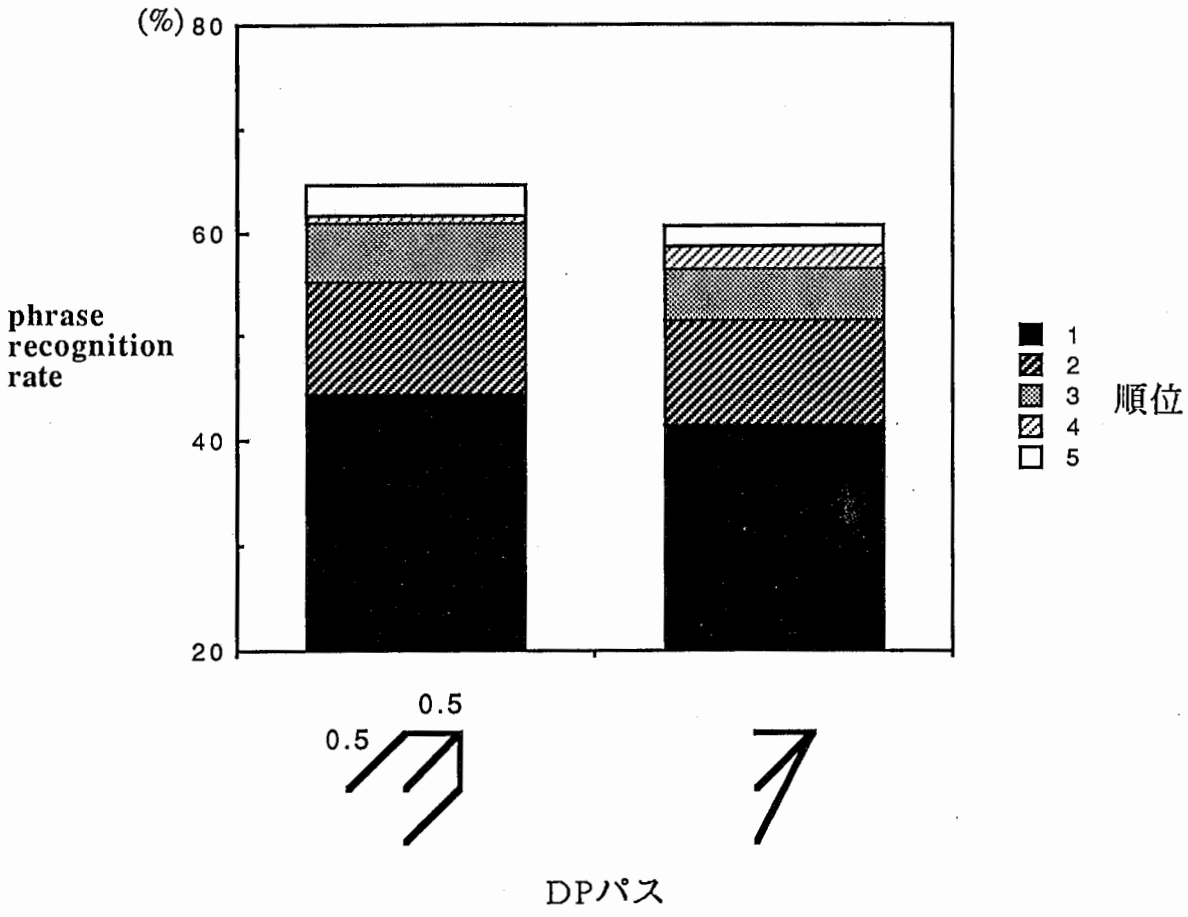
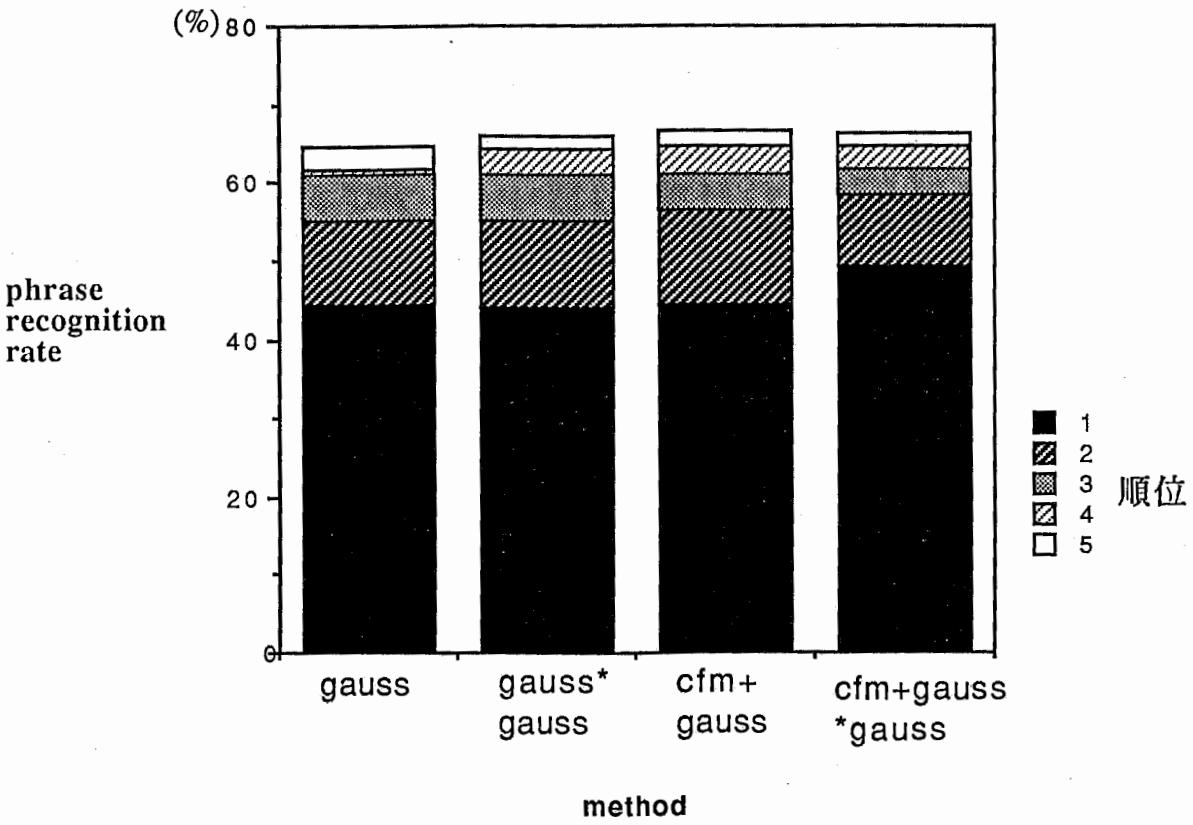


図5-6 DPパスによる文節認識率(1)



実験条件
1.CFM　　α =1.　　β =4.　　ξ =1.

図5-7 CFMによる文節認識率

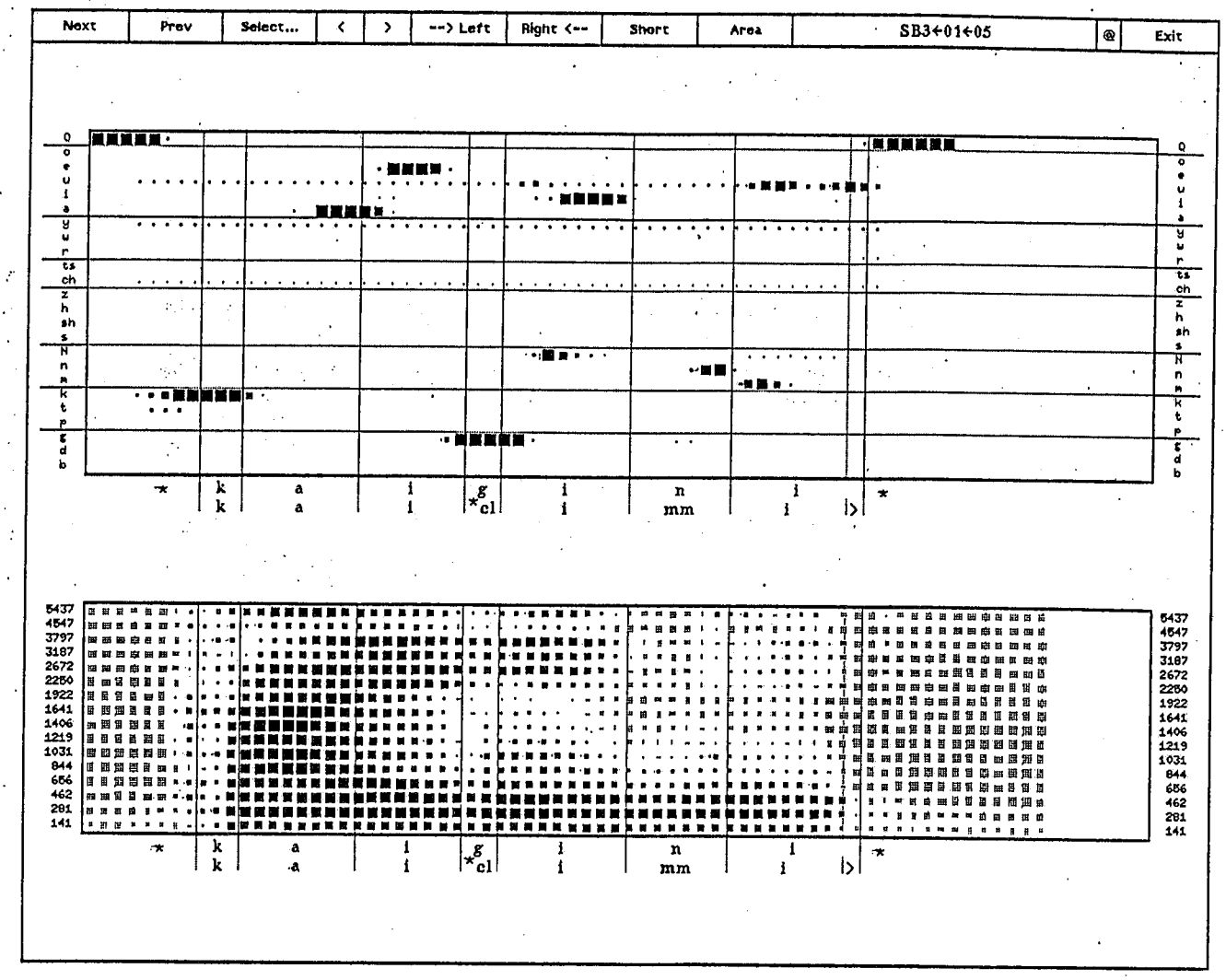


図5-8 MSE によるスポッティング結果の例
(発声は"会議に")

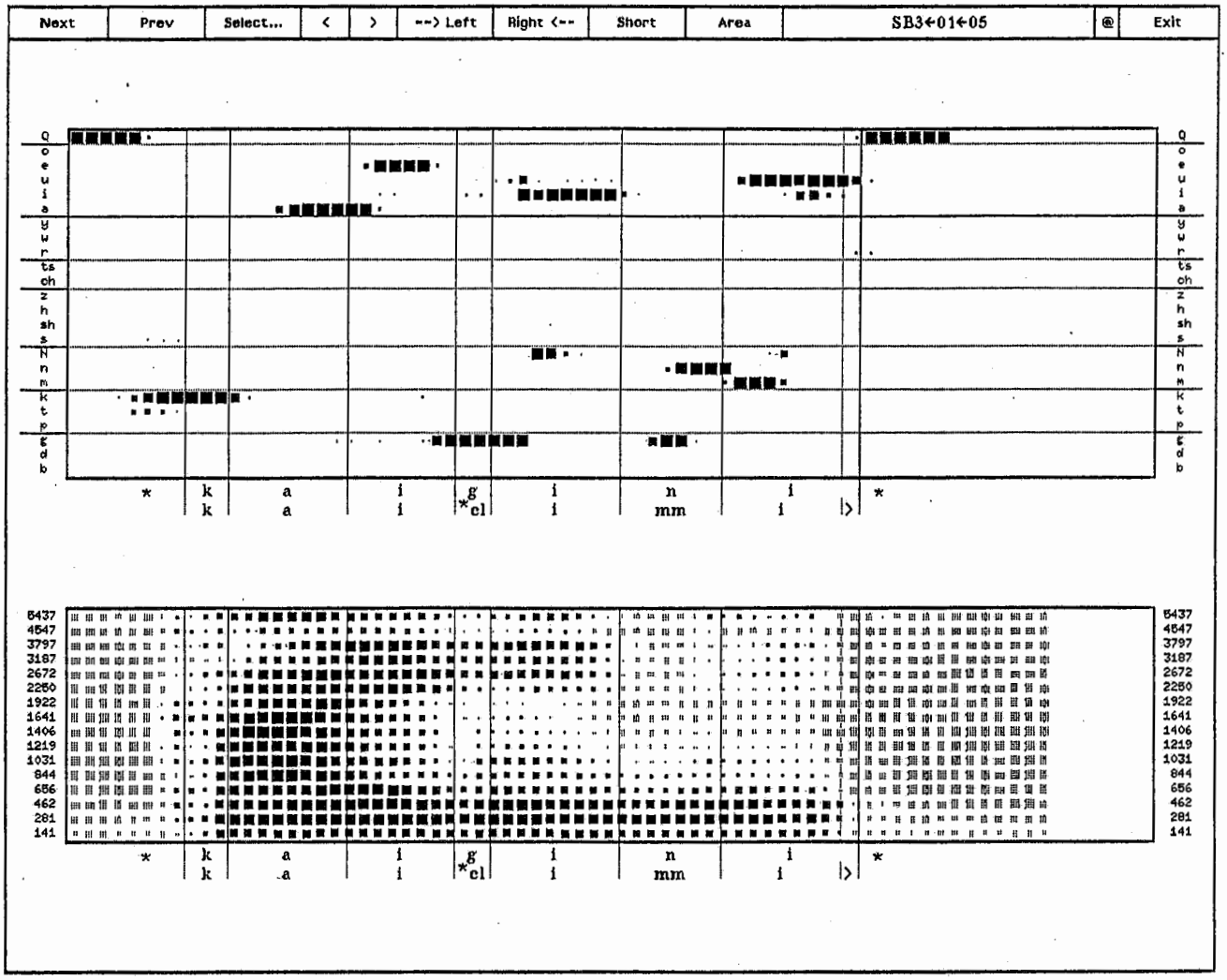


図5-9 CFMによるスポッティング結果の例
(発声は"会議に")

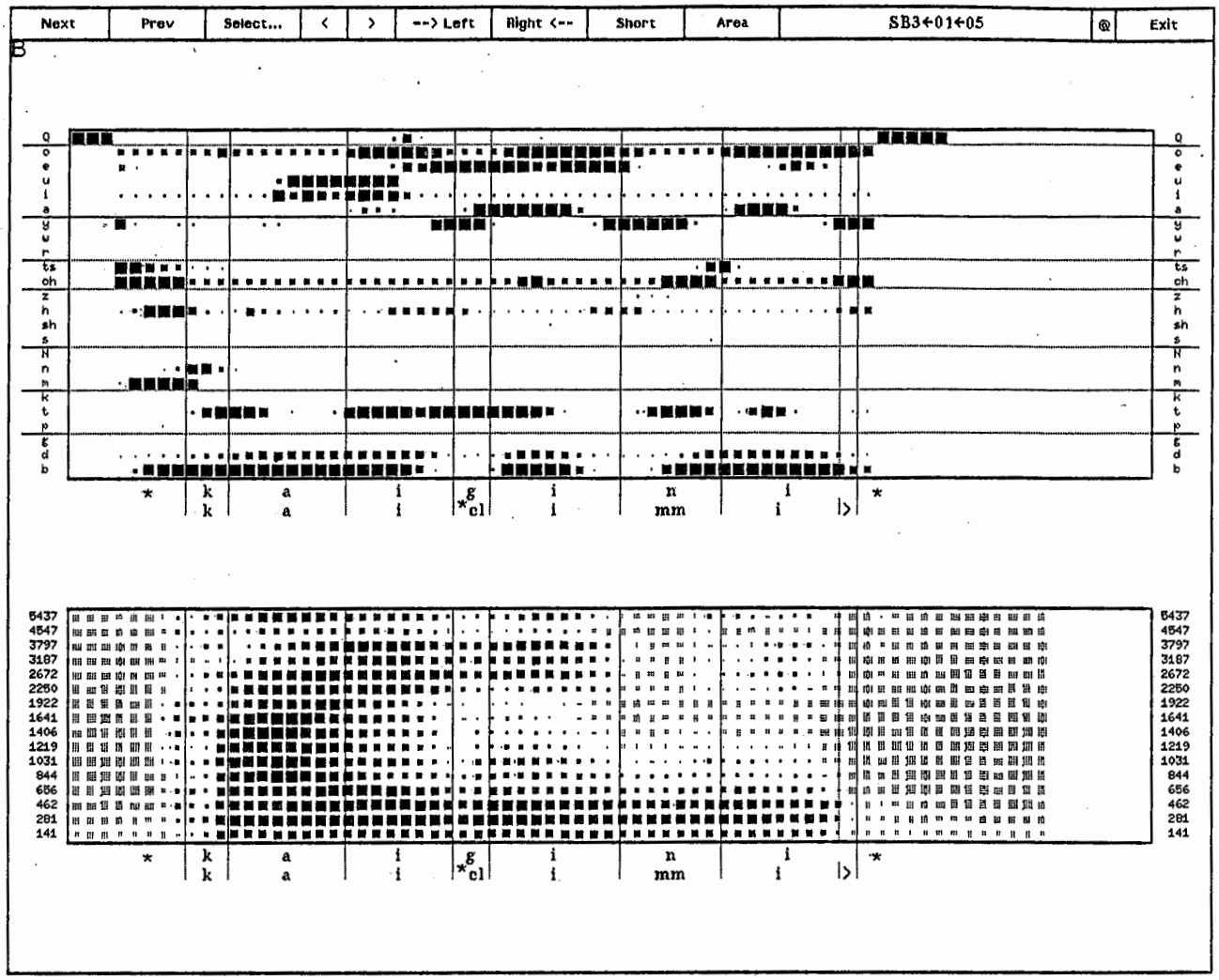


図5-10 弱者救済ネットのスポッティング結果の例
(発声は"会議に")

以下の５．３．２から５．３．８については基本的に以下の実験条件を使用したしかし個々に於てパラメータが異なる場合はその項目毎に条件を添えてある。

実験条件　（１）音韻全域の９６００個のデータ（毎音韻カテゴリ当り４００個）。

（２）出力の補正をする。一定値以下のデータは全て０とした。
（３）ビーム幅（Ｂ）はすべて１００。
（４）ｐｔｋの前のｃｌを考慮。
（５）Ｓｉｌｅｎｃｅ／Ｓｐｅｅｃｈネットあり。
（６）継続時間長制御はガウス自乗を用いる。
（７）出力の補正を行わない。

５．３．２　学習データによる認識率の変化

学習データを変えた場合に認識率がどのように変わるかを調べた。

［１］

実験条件　（１）出力の補正を行わない

５．２．４で示した（１）～（３）のデータについて実験を行った。この結果を図５－１１に示す。図から音韻全域について学習させたものが最も認識率がよいことがわかる。

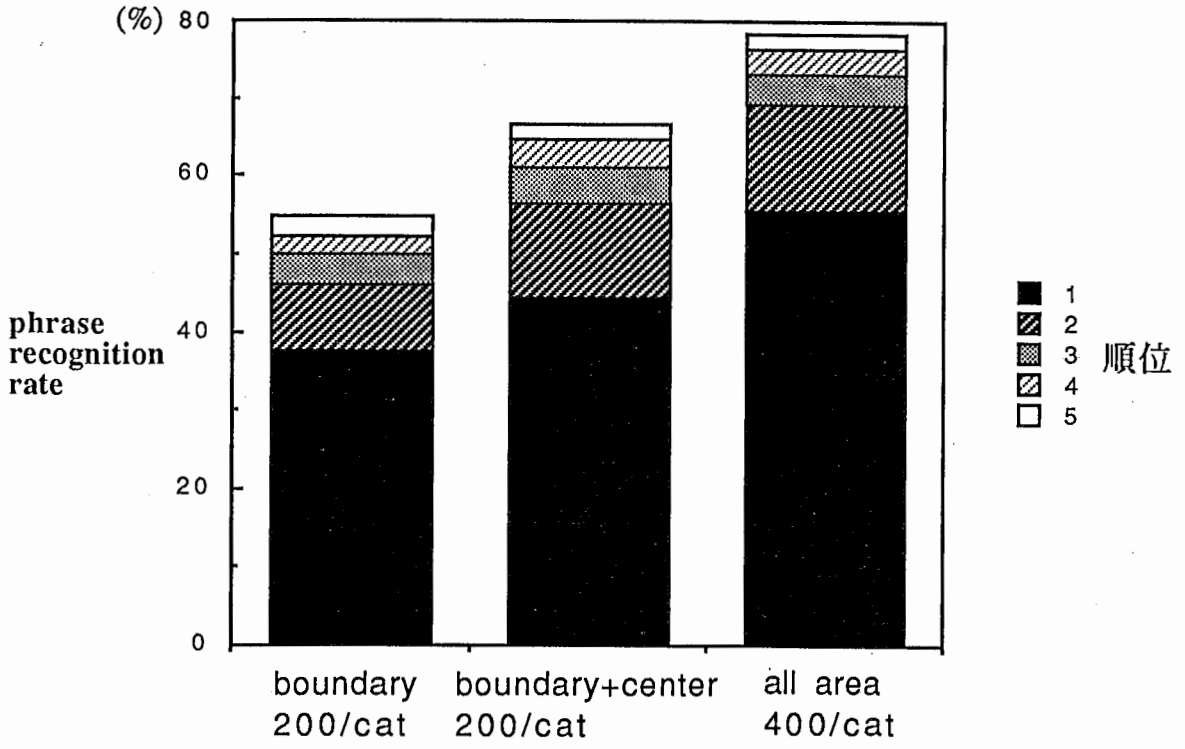


図5-11 学習データによる文節認識率(1)

［２］

実験条件　（１）出力値からオフセットを引く補正を行った

（２）ｐｔｋの前のｃｌを考慮しない

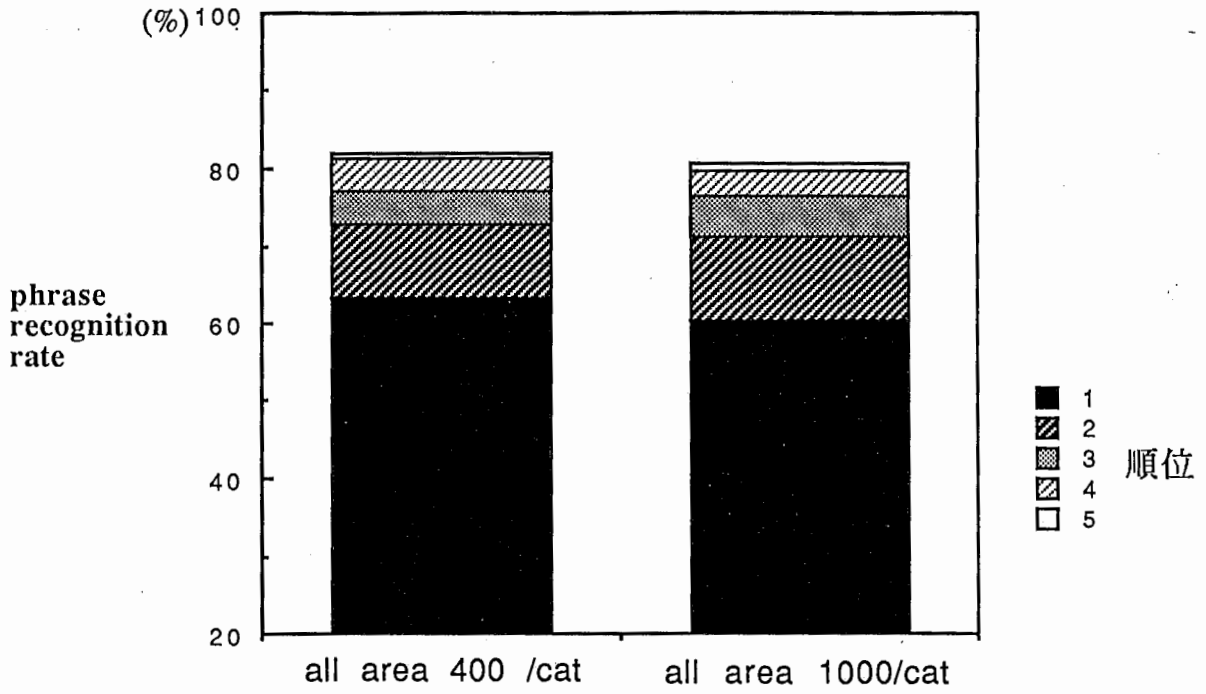
５．２．４で示した（３）と（４）について実験を行った。この結果を図５－１２に示す。この結果は学習の回数について最適化を行っていないので、正確な議論はできない。しかし、傾向として学習サンプルを増やしても認識率の向上にはある限界があることが予想される。

５．３．３　音韻認識率と文節認識率のＴＤＮＮの学習回数による関係

実験条件　（１）音韻境界のデータ（子音）＋音韻中心データ（母音＋Ｎ）。

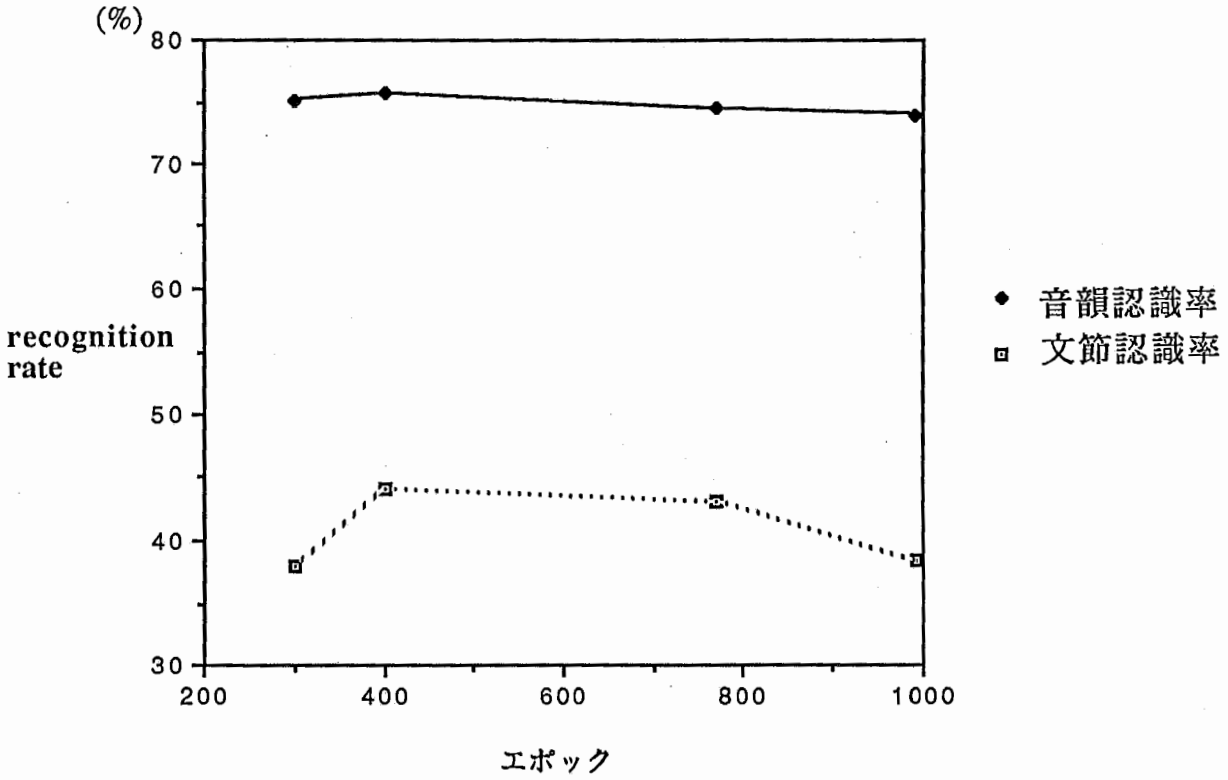
（２）音韻認識には９６００個の音韻全域データを用いている。

この実験はＴＤＮＮの学習回数（エポック）を変化させた時に音韻認識率と文節認識率がどのように変わるかを調べたものである。実験結果を図５－１３に示す。この図から音韻認識率と文節認識率には、極めて強い関係があることがわかる。



認識条件
1.学習エポック 30,20
2.ptkの前のclに対する処理を行なわない

図5-12 学習データによる文節認識率(2)



実験条件
学習　音韻境界+音韻中心(母音+N)
認識　音韻全体
どちらも偶数単語より得た

図5-13 音韻認識率と文節認識率の学習繰り返し回数(エッポク)との関係

５．３．４　ｐｔｋの前の処理について

この実験ではｐｔｋの前のｃｌの処理がどの程度有効であるかを調べたものである。この結果を図５－１４に示す。この結果からｐｔｋの前のｃｌに対する処理があまり有効的でないことがわかる。

５．３．５　出力の補正に対する認識率

実験条件　（１）ｐｔｋの前のｃｌを考慮しない

この実験では５．２．５で示したスポッティングの出力値の補正について調べた。この補正はＤｙｎｅｔを使用する場合には必要である。この結果を図５－１５に示す。この結果からスポッティング結果から各音韻ごとにオフセット値を引いたものが最も認識率がよいことが確認された。

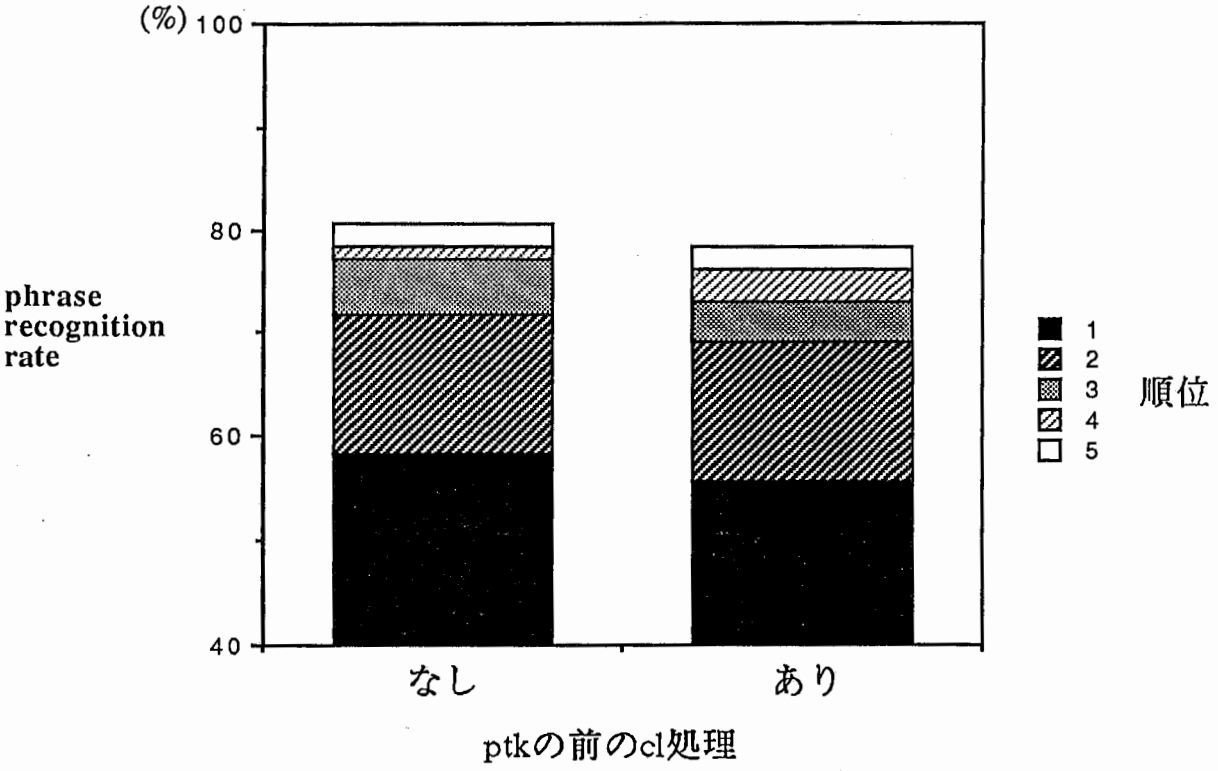


図5-14 ptkの前部にあるcl処理に対する文節認識率

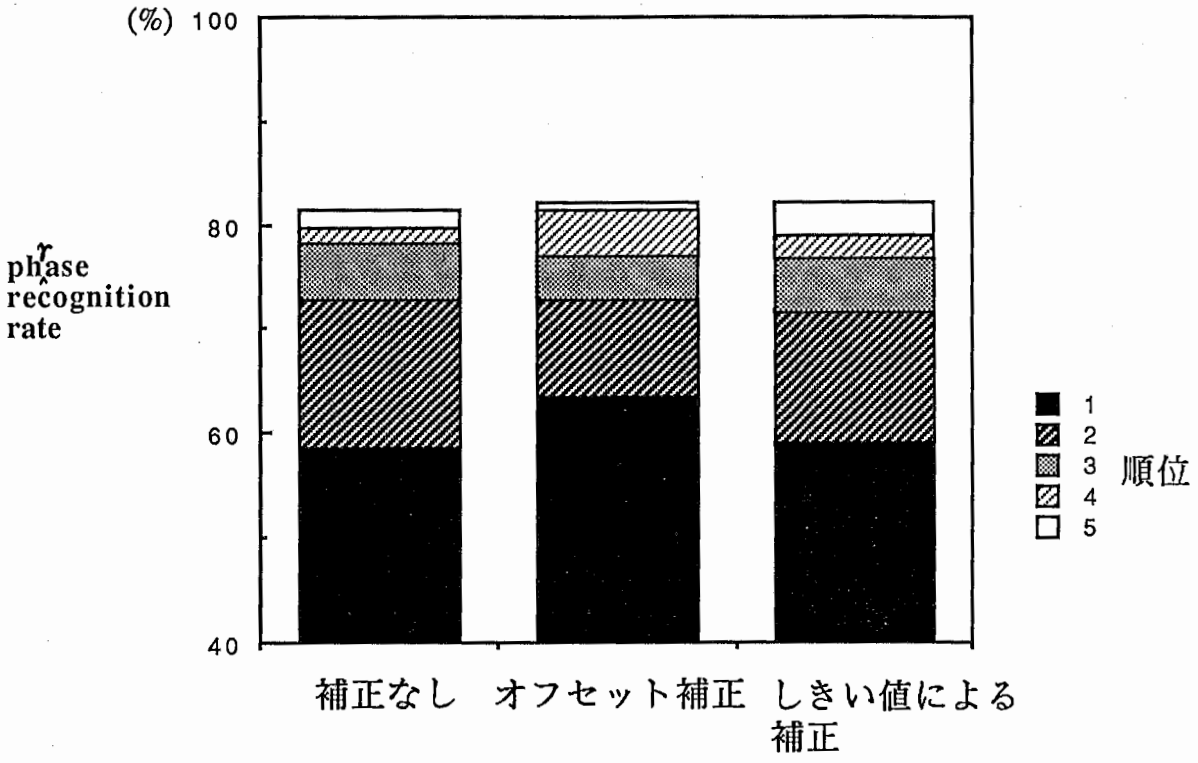


図5-15 スポッティング出力の補正方法による文節認識率の変化

５．３．６　Ｓｐｅｅｃｈ／Ｓｉｌｅｎｃｅネットの必要性について

実験条件　（１）出力からオフセット値を引く補正を行った
（２）ｐｔｋの前のｃｌを考慮しない

この実験では図２－４で示したＳｐｅｅｃｈ／Ｓｉｌｅｎｃｅネットが有効であるかどうかを調べた。この結果を図５－１６に示す。この結果からＳｐｅｅｃｈ／Ｓｉｌｅｎｃｅネットはあまり有効でないが処理速度の向上のためには必要である。

５．３．７　ＤＰパスによる検討

実験条件　（１）出力からオフセット値を引く補正を行った
（２）ｐｔｋの前のｃｌを考慮しない

この実験はＤＰパスによって認識率がどのように変化するかを調べたものである。この結果を図５－１７に示す。この結果から５．２．２で示したＤＰパスのうち、（１），（２）がよいことがわかる。しかし、（１）は文節の最後に音韻の挿入が多い傾向がある。

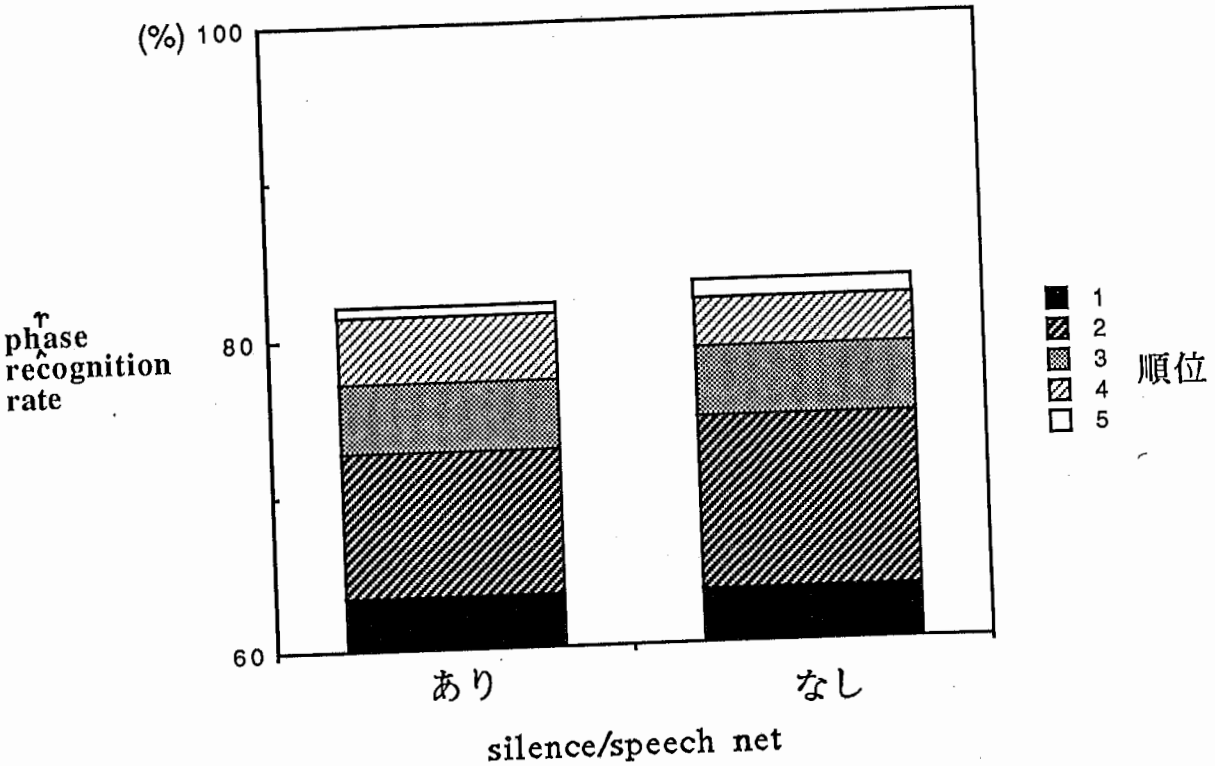


図5-16 Speech/Silenceネットの有無に対する文節認識率

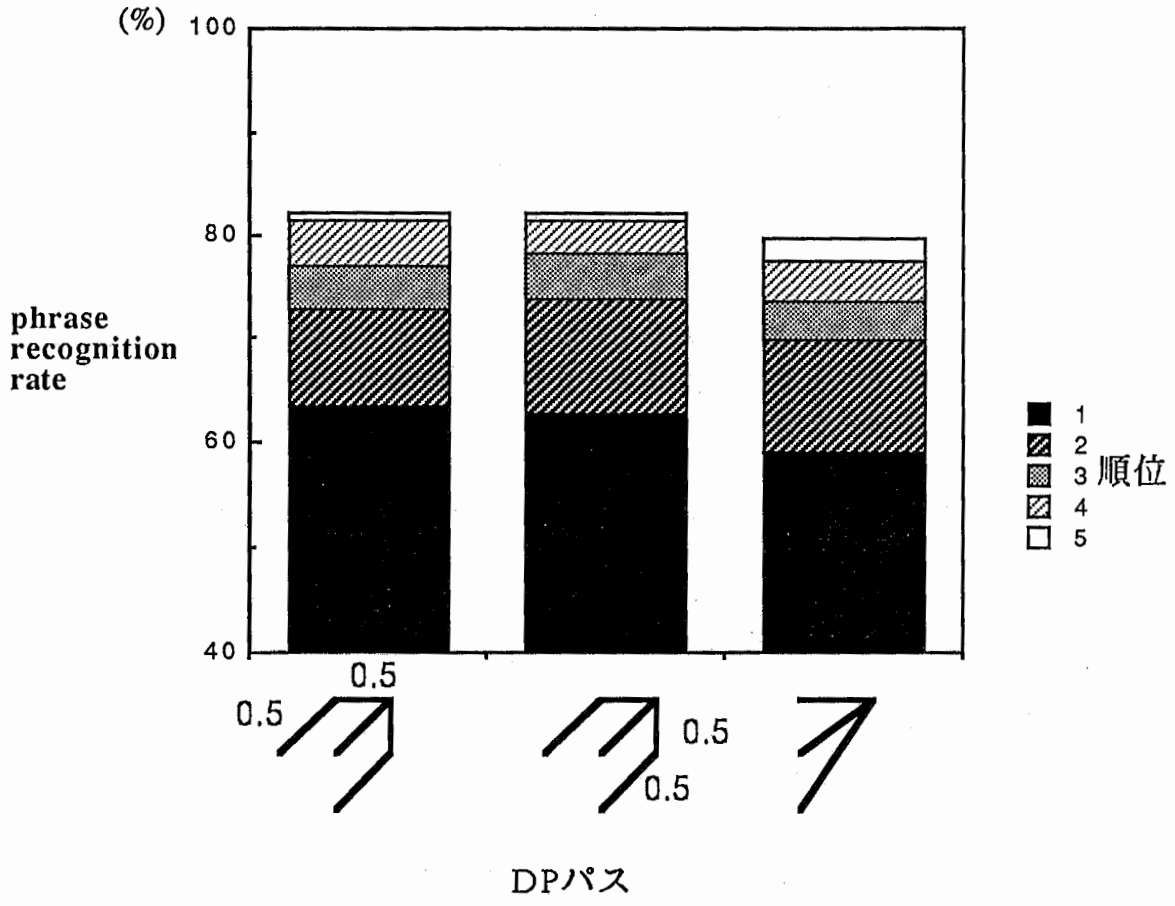


図5-17 DPパスによる文節認識率(2)

6．　解析と検討

6．1．　認識誤りの原因について

以下に5．3．2において認識率が最高になるときの誤認識の原因を影響の大きいものから順に示す。

（1）置換誤り
（2）脱落誤り
（3）挿入誤り
（4）Ｄｙｎｅｔのオフセットによる影響
（5）Ｓｐｅｅｃｈ／Ｓｉｌｅｎｃｅネットの影響
（6）文節終了判定の誤り

この原因は1位と認識されなかった１１６文節に対して検討したものである。この原因のうちｏｆｆｓｅｔ、ｓｐｅｅｃｈ／ｓｉｌｅｎｃｅネットについての考察は前節の5．3．5、5．3．6にで述べた。

誤認識の最も大きな原因は音韻の脱落、置換であると考えられる。特に母音とＮはｕに置換され易く、これが大きな認識率低下の原因となっている。これは24音韻のＴＤＮＮそのものの問題である。

7．　今後の課題

7．1．　現在のシステムの最適化

現在構築したシステムを最適にするためには次に挙げるいくつかの項目について改良する必要がある。

7．1．2．　学習サンプルの選定法

認識率は学習サンプルの選択によって大幅に変化するので学習サンプルについての最適化が必要である。具体的には音韻中の学習サンプルの場所の決定、一つの音韻カテゴリのサンプル数などを最適に決定する必要がある。

7．1．3．　文節認識の終了時の判別方法の修正

現在文節の終了の判別では、最適パスのフレームがラベルファイルの終了フレームからある閾値以内にあるかないかを調べている。しかし、この方法では文節末の音韻が長母音の場合、文節が終了したと判別できなくなる。これを解決するためには文節頭と文節尾に無音区間の音韻を付加した辞書を用いてＤＰを行う必

要がある。この場合には無音区間と他の音韻とがマッチングしないようにする必要がある。

7．1．4．　flooringの最適化

flooringの値を修正する場合にはDynetの出力値を補正する必要がる。

7．1．5　学習回数の最適化

文節の認識率はTDNNの学習回数によって大幅に変化するため、この項目の最適化も必要である。

7．1．6　音韻継続時間長の再統計

現在のTDNN－LRで用いている音韻継続長はHMM用に作成したものであるのでTDNN用に修正する必要がある。

7．2　根本的な問題

いままでの実験からTDNNの学習そのものに問題があることが確認された。このことから学習の根本的な改良が必要であると考えられる。この改良としては以下の2点が挙げられる。

7．2．1　出力値のファジー化

現在の出力は0，1の2値になる傾向が強い。このため致命的な脱落、置換がきわめて多い。これを救うためにはTDNNの出力を現在よりファジー的にする必要がある。

7．2．2　逐次的学習

学習が逐次的に無限に可能であれば、コンピュータのメモリによる制限などを取り除くことが可能である。学習が逐次的に可能であれば、ネット自身がデータにあわせて適応化することが可能である。

謝辞

本研究の機会を与えて下さった樽松明社長に感謝致します。また、有益なアドバイスを頂いた北研二氏をはじめとするATR自動翻訳電話研究所音声情報研究室およびデータ処理研究室の皆様に感謝致します。

参考文献

[1]宮武、沢井、鹿野：”時間遅れニューラルネットワークを用いた音韻スッポテ

ィング法”，電子情報通信学会春期全国大会（１９８９年３月）

[2]鹿野　他：”ニューラルネットワークの音声情報処理への応用”，ＡＴＲ Ｔｅｃｈｎｉｃａｌ　Ｒｅｐｏｒｔ　ＴＲ－Ｉ－００６３（１９８８．１２）

[3]北、川端、斉藤：”ＨＭＭ音韻認識と予測ＬＲパーザを用いた文節認識”，信学技報ＳＰ８８－８８（昭和６３年１０月）

[4]P.Haffner : " DyNet,a Fast Program for Learning in Neural Networks ", ATR Technical Report TR-I-0059(1988.11)

[5]J.B. Hampshire II A. H. Waibel:"A Novel Objective Function for Improved Phoneme Recognition Using Time Delay Neural Networks", International Joint Conference on Neural Networks, Vol.I, pp235-241 (1989.6).