TR-I-0211

TDNN-HMM-LR による文節音声認識

TDNN-HMM-LR Applied to Phrase Recognition

ShiNya YAMAFUKU & Masahide SUGIYAMA

内容梗概

TDNN (Time Delay Neural Network) は高い音素識別能力を持つことが示されている。その後 TDNN-LR として、文節認識系が構築されたが、確率的なマルコフモデルである HMM-LR には文節認識率においては十分な性能は実現できていない。これは HMM-LR において行なわれている、全探索に対して、音声時系列と音素記号系列との記号列間のマッチングになっていることや、時間情報（音素継続時間の情報など）が十分に表現されていないことなどが原因と考えられている。ここでは TDNN-LR では DP が用いられているのに対して、LR と HMM を用いて接続することを検討する。この音素認識に対しての性能はすでに報告したので、ここでは文節認識に対する性能の評価について述べることにし、その性能、問題点などを明らかにする。

ATR Interpreting Telephony Research Labs.

目次

図目次

表目次

1 はじめに

TDNN (Time Delay Neural Network) は高い音素識別能力を持つことが示されている [1]。その後 TDNN-LR として、文節認識系が構築されたが、確率的なマルコフモデルである HMM-LR には文節認識率においては十分な性能は実現できていない。これは HMM-LR において行なわれている、全探索に対して、音声時系列と音素記号系列との記号列間のマッチングになっていることや、時間情報（音素継続時間の情報など）が十分に表現されていないことなどが原因と考えられている。ここでは TDNN-LR では DP が用いられているのに対して、LR と HMM を用いて接続することを検討する。この音素認識に対しての性能はすでに報告したので [2]、ここでは文節認識に対する性能の評価について述べることにし、その性能、問題点などを明らかにする。本検討は以下のような研究の位置付けとしてとらえることもできる。

- segment-based HMM recognizer の性能の把握

  近年、segment-based の認識系が盛んに研究されている。TDNN を segment-based 特徴抽出器としてとらえ、その性能を把握する。

- TDNN-LR の性能改善

  TDNN, LR, その interface のいずれに問題があるのかを明らかにする。

- HMM と NN との融合 [4],[5]

HMM と NN との融合についての最近の研究は文献 [2] において概観している。

2 準備

2.1 TDNN について

TDNN は 23 音素及び無音を認識するために、図 1のような modular 型構造を用いることにした。

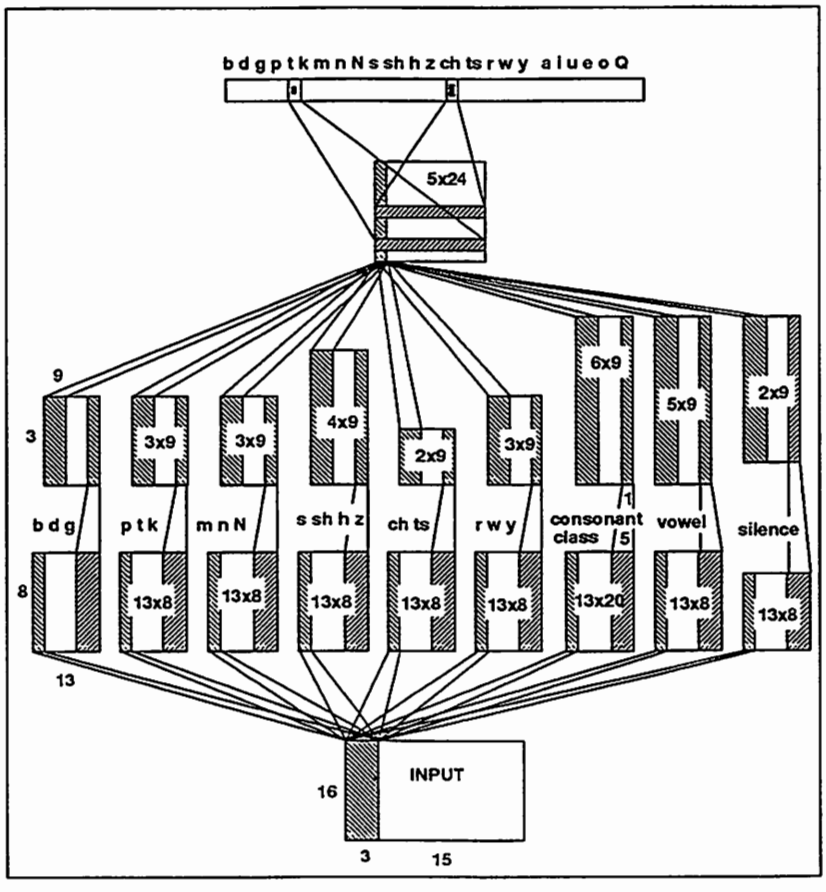

図 1: The Modular TDNN 18 Consonant, 5 Vowel and Silence Recognition Architecture

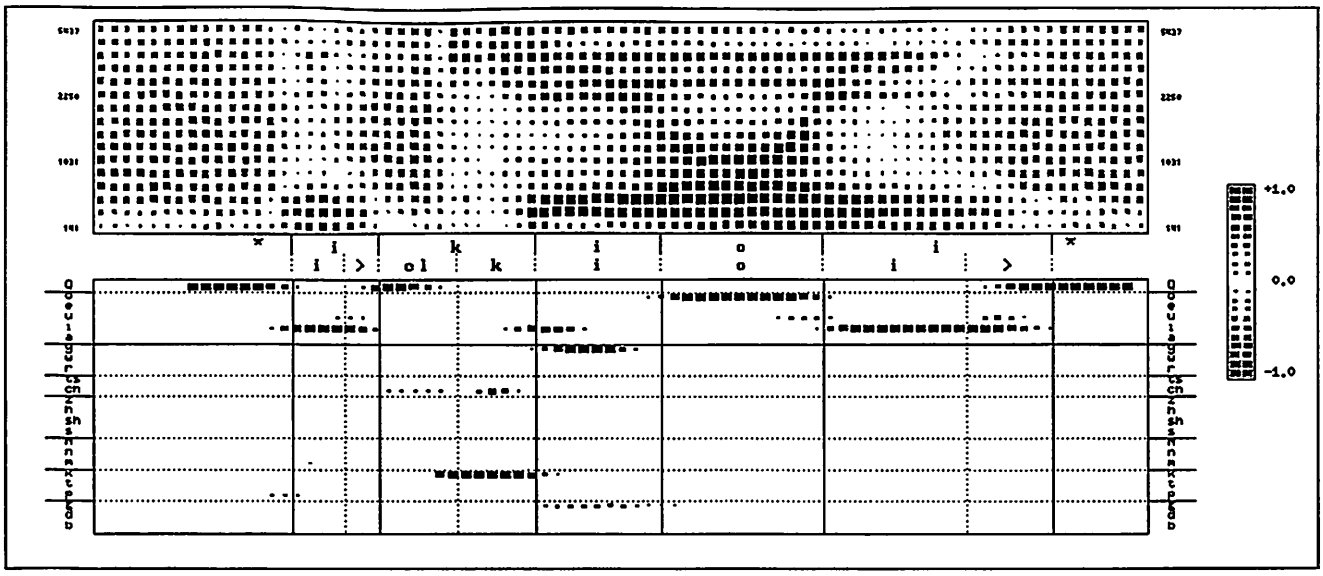

図 2: The TDNN output firing pattern for a word */ikioi/*

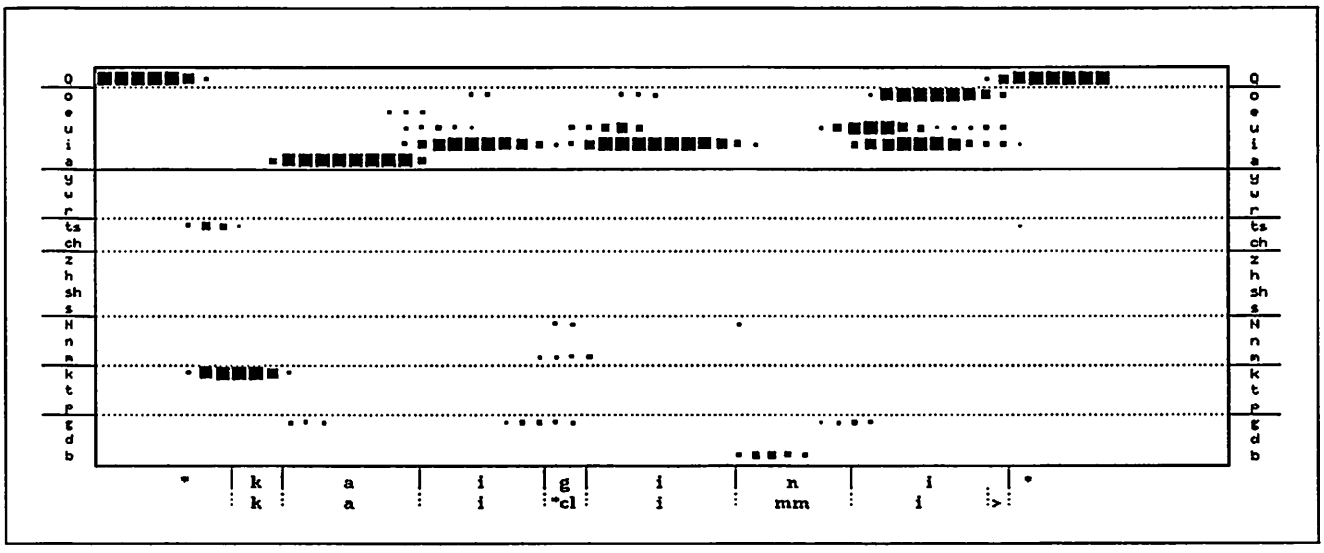

図 3: The TDNN output firing pattern for a phrase */kaigini/*

入力単語に対して、TDNN の出力パターンを図 2 に示す。また、文節音声に対する TDNN の出力パターンを図 3 に示す。付録 A に認識用の文節音声の TDNN 出力パターンを 2 文章分だけ示す。

2.2 VQ について

従来の検討結果 [2] を踏まえてここでは VQ 符号帳の大きさを 256 とした。符号帳の大きさと歪みとの関係を表 1 に示す。

表 1: Decreasing VQ distortion

codebook size	distortion
2	0.7858
4	0.5379
8	0.4886
16	0.3550
32	0.2264
64	0.1675
128	0.1250
256	0.0971

2.3 HMM について

母音・無音は 1 状態であり、子音は 3 状態を用いている。HMM の構造を図 4に示す。従来の HMM-LR においては 24 音素の認識のために表 2に示すような 69 個の HMM を用いてきた。しかしながら、TDNN-HMM-LR においては簡単のために表 3に示すような 24 個のモデルのみ用いることにした。この時の $/b, d, g/$ 音素認識率を表 4に示す。TDNN-HMM は TDNN と HMM との中間の認識率を与えているので、この性能が TDNN-HMM-LR においても維持されることを想定している。

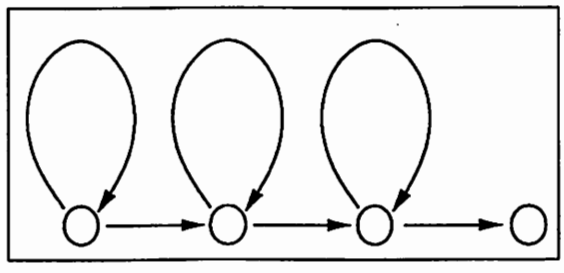

図 4: HMM の構造

表 2: A Set of the Traditional HMMs for HMM-LR

s sh h z ch1 ch ts1 ts p1 p t1 t k1 k b1 b d1 d g ng m n N N3 r1 r w y
a a3 i i3 Ui u u3 Uu e e3 o o3
aa aa3 ii ii3 uu uu3 ee ee3 oo oo3 ei ei3 ou ou3
sy hy zy cy py ky by gy ngy my ny ry
Q Q1 Q2

表 3: A Set of the TDNN-HMMs

b, d, g, p, t, k, m, n, N, s, sh, h, z, ch, ts, r, w, y
a, i, u, e, o
Q

表 4: Comparison of recognition rate

HDVQ: Hard-VQ
FZVQ: Fuzzy-VQ

method	b	d	g	average rate
HDVQ-256	97.7	96.6	96.8	97.0
FZVQ-256	98.6	97.2	96.0	97.2
HMM only	95.6	97.2	94.4	95.6
TDNN	98.2	98.3	99.6	98.7

2.4 Grammar について

一般向け文法を用いて、HMM-LR を動作させた。

2.5 Duration の学習について

継続時間長は作成した TDNN-HMM に基づいて学習を行なっている。ここで、通常の HMM-LR において継続時間長の種類で異なる HMM として扱っているものがあるが、本 TDNN-HMM-LR においては同一の継続時間長で処理を行なっている。継続時間長の学習は DSB を用いて行なった。

2.6 音声資料

VQ 符号帳の作成には 216 音素バランス単語を用いた。さらに、HMM の学習には 5240 単語の偶数番目を用いた。文節音声を用いて評価を行なった。音声の分析条件を表 5に示す。

表 5: Speech Analysis Specification in TDNN Recognition System

speaker	1 male
sampling frequency	12kHz
windowing	256 point Hamming window
frame rate	10ms
feature parameter	16 channel FFT mel-scaled spectrum 15 frames
power normalization	normalized to lie between -1.0 and +1.0 with the average at 0.0

文章毎に一つのファイルに格納されている、文節音声資料に対して TDNN の出力パターンを計算する。この時、ATR-TDNN software は音素ラベル情報に基づき文節音声毎の切り出しを行ない、一つのファイルに格納している。これを HMM-LR 認識系に入力するためにそのファイルにおける音声の開始・終了を指定しなければならない。ここでは以下のように指定した。54.0 ms の数値は HMM-LR の認識系における音声切り出し位置が 0 ms になるように設定した。

$$start = 54.0\ (ms)$$
$$end = 54.0 + end\_sentence - start\_sentence\ (ms)$$

3 TDNN-HMM-LR を用いた文節音声認識

全体の概念図を図 5に示す。入力文節音声は TDNN を用いて segment 単位に特徴抽出され、24 次元のベクトルに変換される。このベクトルはあらかじめ作成されているベクトル符号帳 (256 符号) を用いて量子化され、記号列に変換される。この記号列はあらかじめ学習されている離散型 HMM へ入力される。音素 HMM は LR パーザーによって駆動され

て音素毎に認識されその尤度が算出される。VQ 部分については連続 HMM を用いる場合には必要はない。ここでの実験には pow flag を用い、duration weight は 4 に設定している。

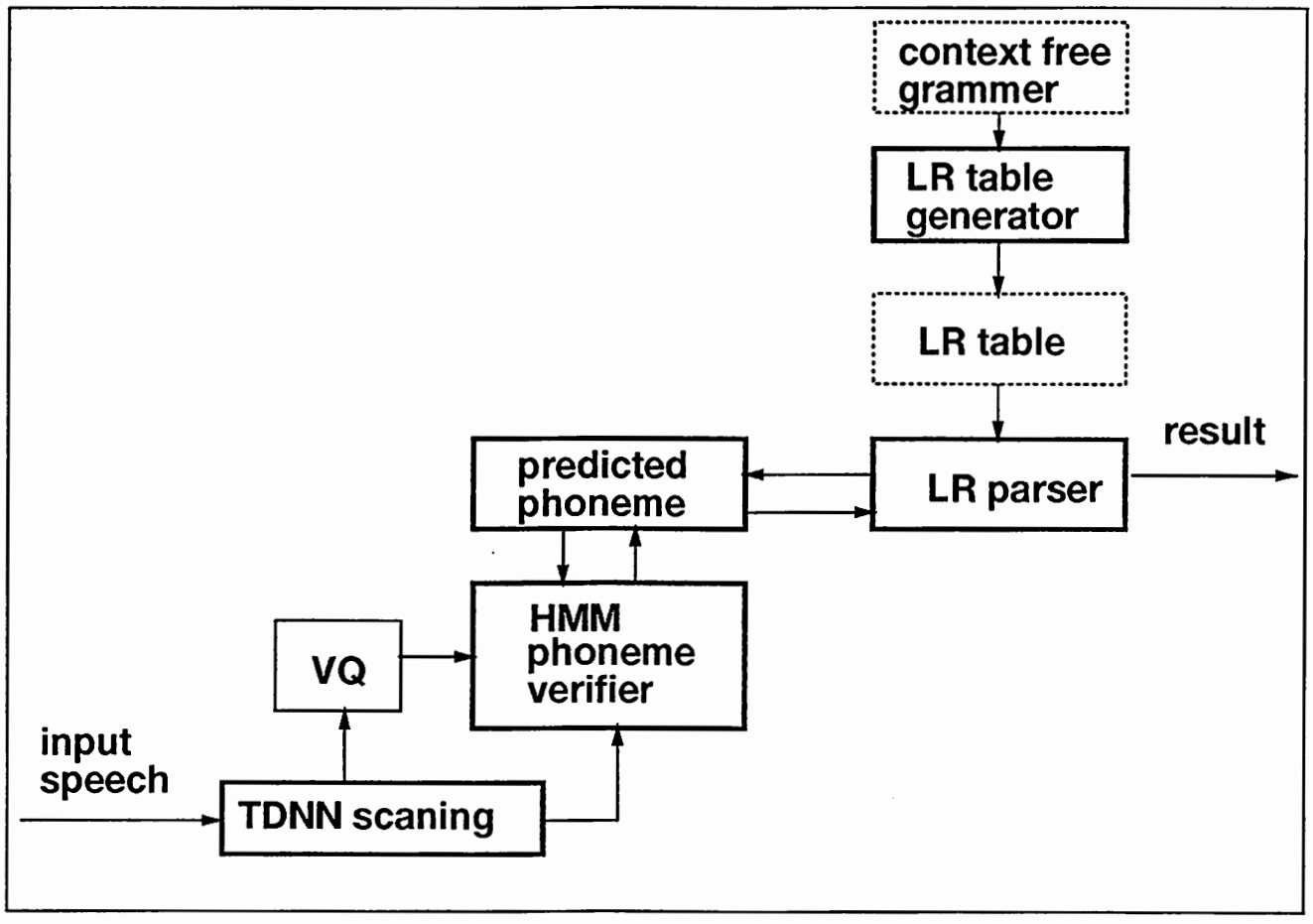

図 5: TDNN-HMM-LR の概念図

認識性能を以下の表 6に示す。これらの TDNN-LR および HMM-LR の結果は文献 [3] から引用している。認識結果を比較してみると第 5 位までの認識率が他の方法に比べて低い。これは改善されると、HMM-LR の認識率にせまるものと思われる。TDNN-HMM-LR においては扱いの単純化のために 256 個からなる符号帳を用いている。わずかこれだけの個数を用いた認識性能であることを考えると、さらに改善の余地があるように思われる。

表 6: Rates for 279 Japanese Phrase Recognition

rank	phrase recognition rate (%)			
	TDNN-HMM	TDNN	HMM Single VQ	HMM Separate VQ
1	49.5	55.0	72.0	83.2
≤ 2	59.5	70.1	85.3	93.9
≤ 3	63.8	76.6	91.8	96.4
≤ 4	69.5	81.3	94.3	97.5
≤ 5	71.3	82.7	95.3	98.6

4 むすび

音素認識においてその有効性が示された TDNN-HMM 認識法を文節音声に適用した TDNN-HMM-LR について検討した。単純な HMM との結合では認識性能の向上は見られなかった。さらに、TDNN-LR の認識性能にも及ばない結果となっている。文節音声に対する TDNN の出力パターンにおいては発火の見られない音素部分が存在し、これのために認識において reject が発生しているものと思われる。これらを踏まえてさらに HMM との結合方法を検討する必要がある。

今後は以下の項目を検討する。

- TDNN の隠れ層に対する Continuous Mixture Model を用いた評価
- TDNN 出力に対する Fuzzy VQ の効果
- Fuzzy Training TDNN[6] による出力の平滑化の効果

謝辞

本報告は報告者の監督の元で京都工芸繊維大学アルバイト学生山福君によって 1991 年 2 月 18 日から 4 月 5 日になされたものである。報告者の不十分な指導にもかかわらず熱心に本研究を進めて TDNN-HMM-LR の認識系を動作してくれた山福君に感謝します。日頃御指導いただいています嵯峨山室長、また、HMM tool を提供していただいた服部氏、種々の助言をいただいた大倉氏に感謝します。

参考文献

[1] Y.Minami, T.Hanazawa, H.Iwamida, E.McDermott, K.Shikano, S.Katagiri, M.Nakagawa, On the Robustness of HMM and ANN Speech Recognition Algorithms, Proc. of ICSLP90, 31.3, pp.1345-1348 (Nov. 1990).

[2] Alain Biem, Masahide Sugiyama, Study on Combining HMMs and Neural Network Models, TR-I-0174 (Feb. 1991).

[3] Y.Minami, M.Miyatake, H.Sawai, K.Shikano, Continuous Speech Recognition Using TDNN Phoneme Spotting and Generalized LR Parser, Proc. ASJ, 3-1-11, pp.97-98 (Oct. 1989) (in Japanese).

[4] Les T. Niles and H.F. Silverman, Combining Hidden Markov Model and Neural Network Classifiers, Proc. ICASSP90, S8.2, pp.417-420 (1990-04).

[5] N.Morgan and H.Bourland, Continuous Speech Recognition using Multilayer Perceptrons with Hidden Markov Models, Proc. ICASSP90, S8.1, pp.413-416 (1990-04).

[6] Y.Komori, A.Waibel, S.Sagayama, A new Fuzzy Training Method for Phoneme Identification Neural Networks, Proc. of ASJ, 1-5-15, pp.33-34 (Mar. 1991) (in Japanese).

A 文節音声の TDNN 出力パターン

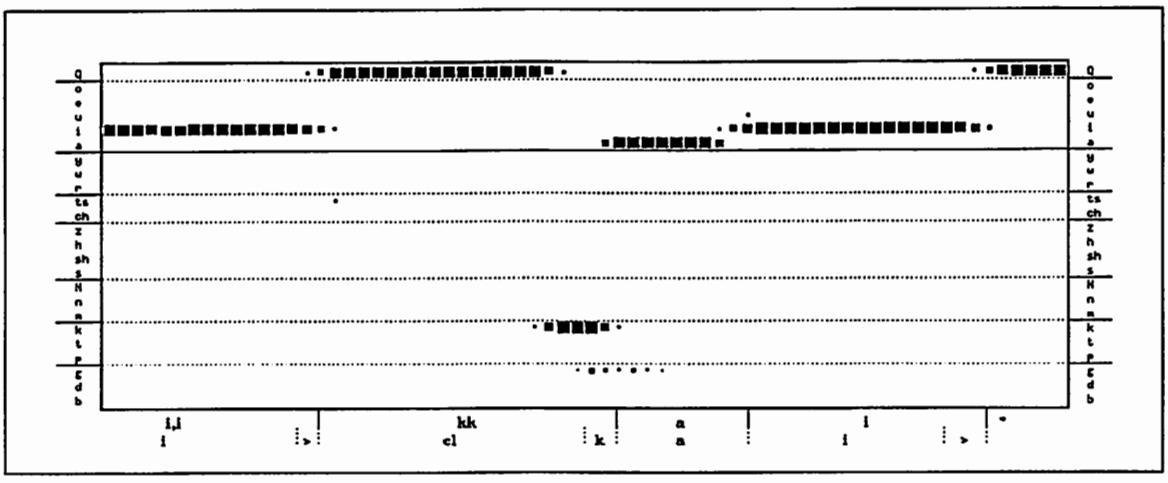

図 6: TDNN の出力パターン (daiikkai)

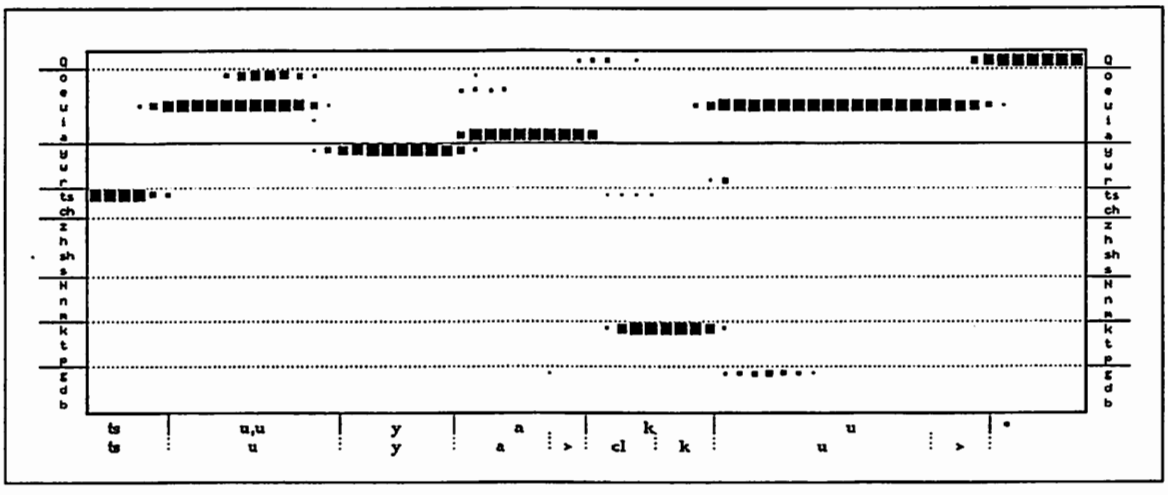

図 7: TDNN の出力パターン (tsuuyaku)

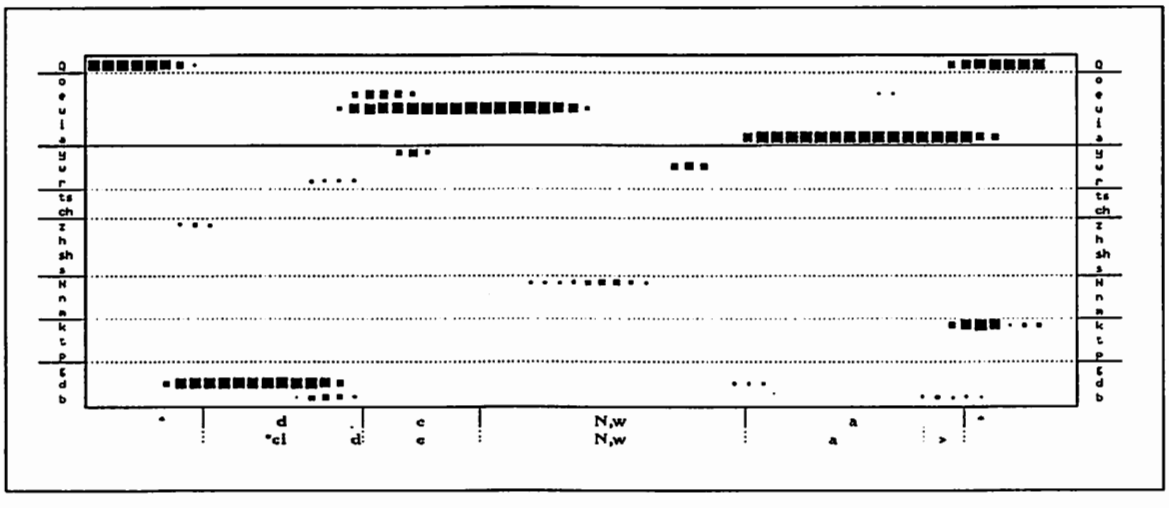

図 8: TDNN の出力パターン (deNwa)

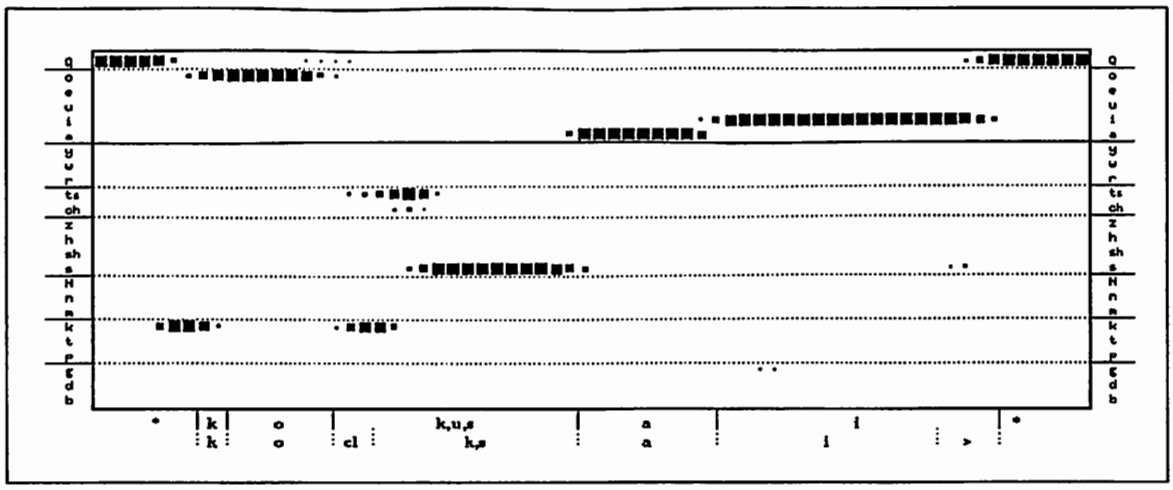

図 9: TDNN の出力パターン (kokusai)

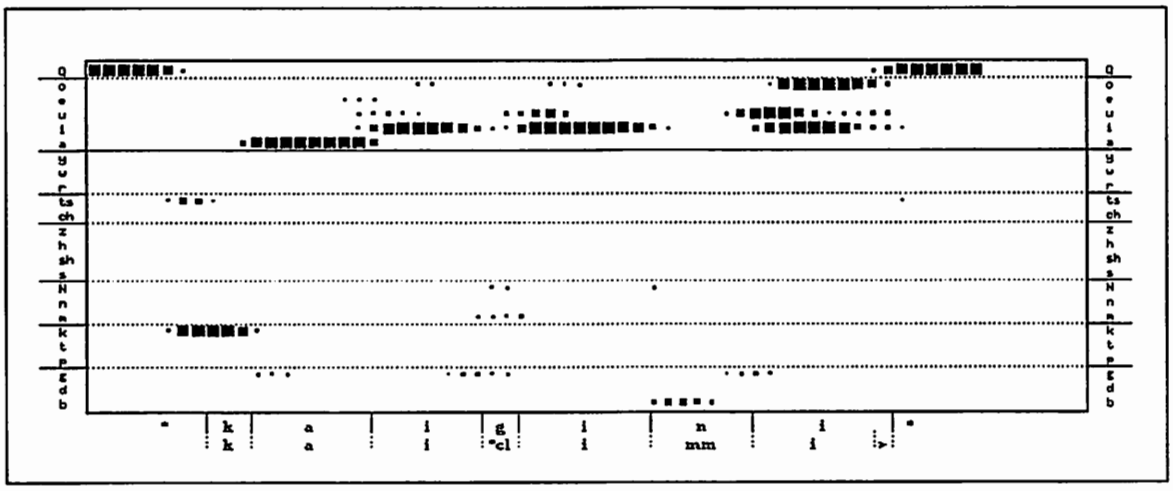

図 10: TDNN の出力パターン (kaigini)

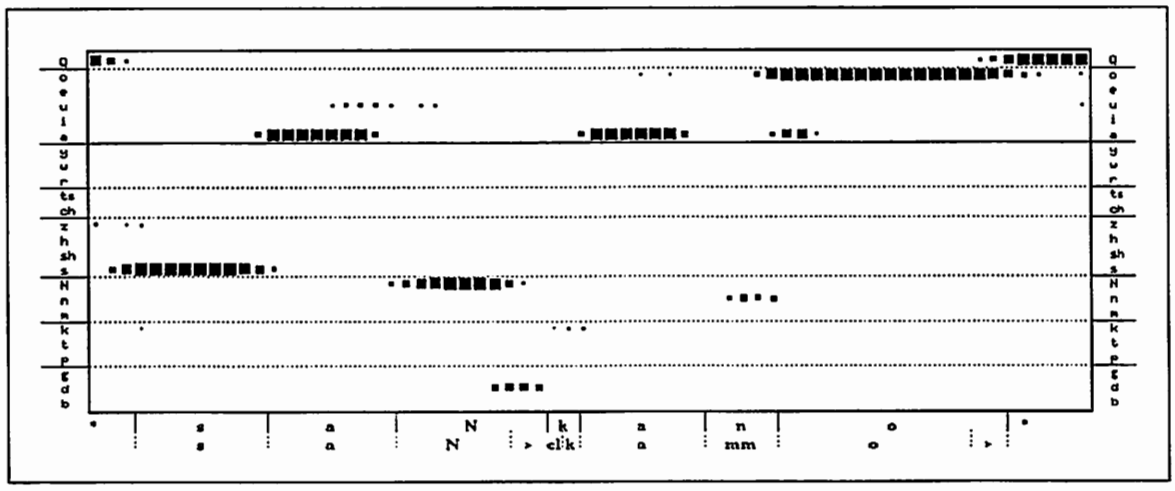

図 11: TDNN の出力パターン (saNkano)

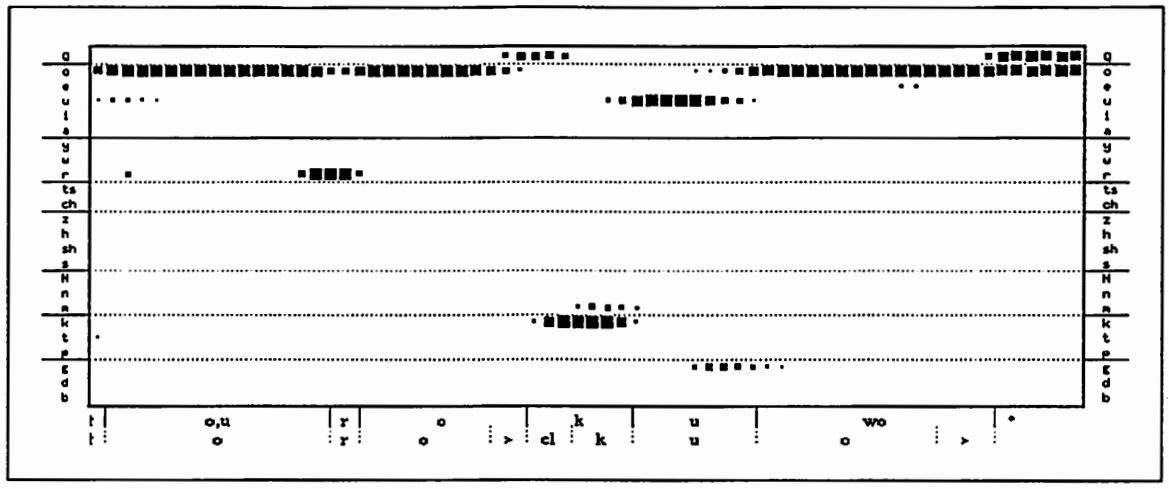

図 12: TDNN の出力パターン (tourokuo)

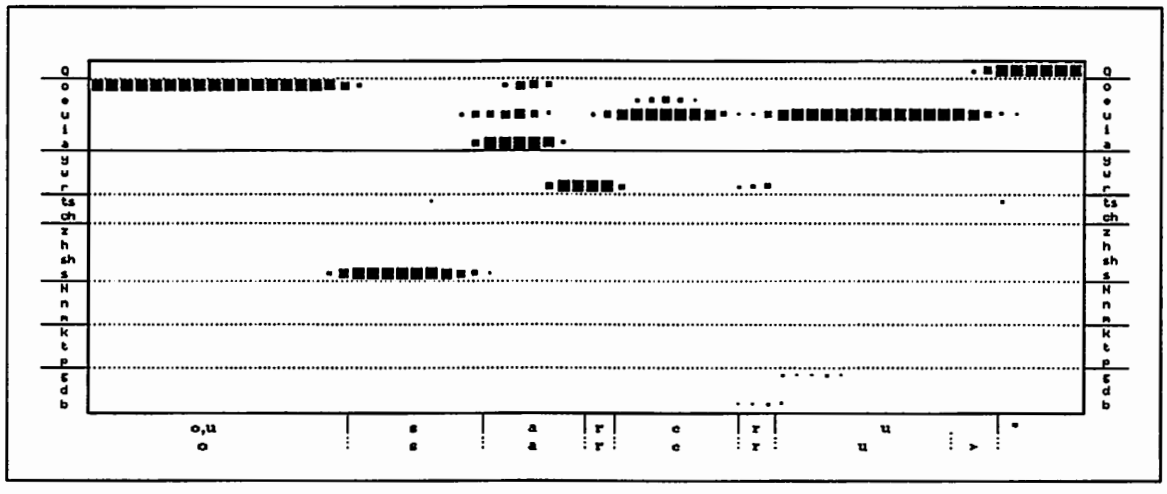

図 13: TDNN の出力パターン (gokibousareru)

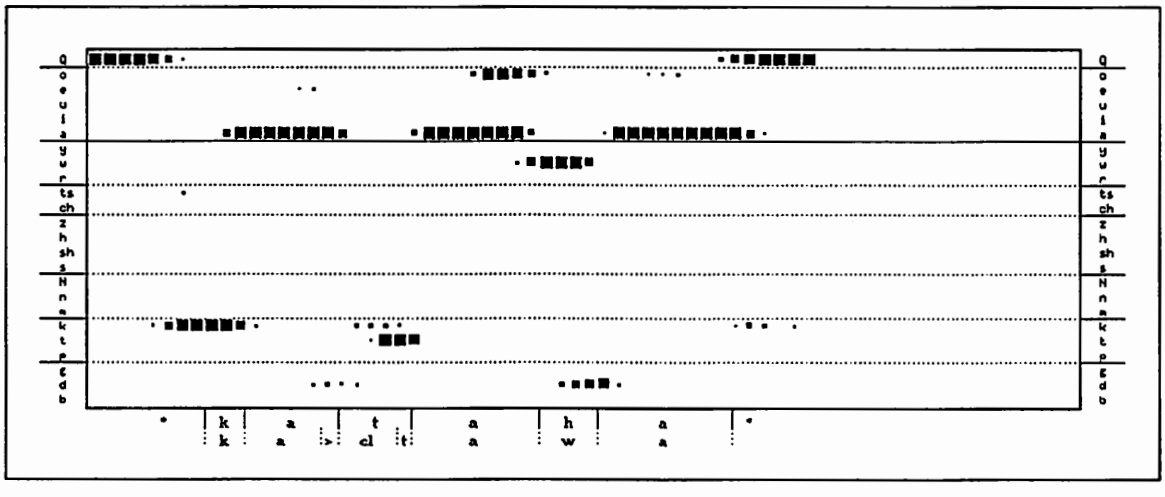

図 14: TDNN の出力パターン (katawa)

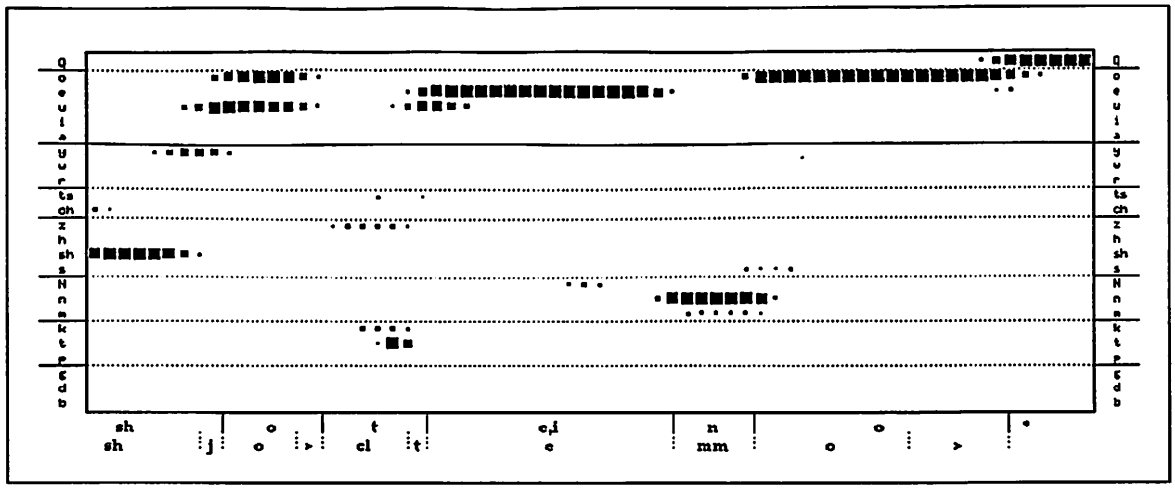

図 15: TDNN の出力パターン (shoteino)

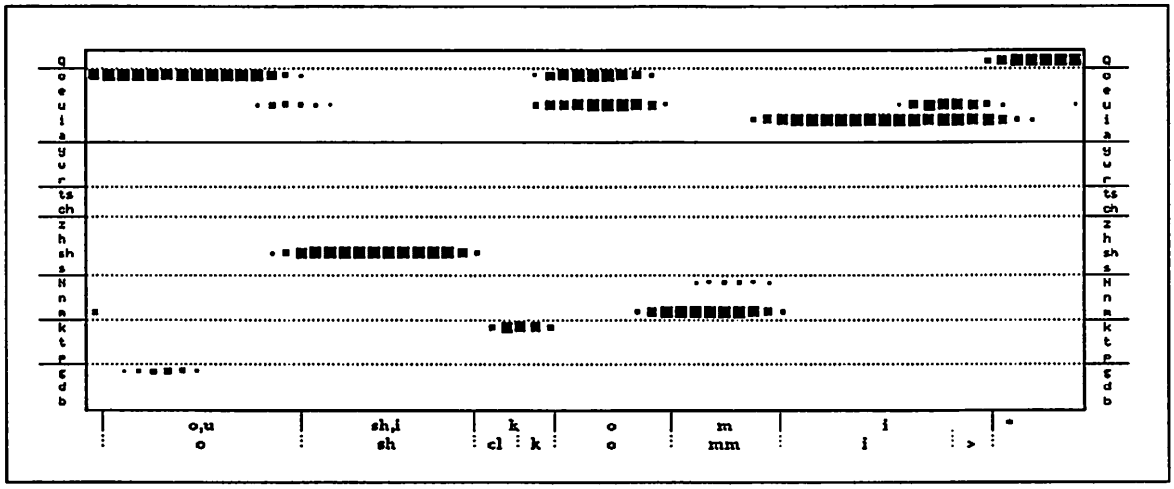

図 16: TDNN の出力パターン (moushikomi)

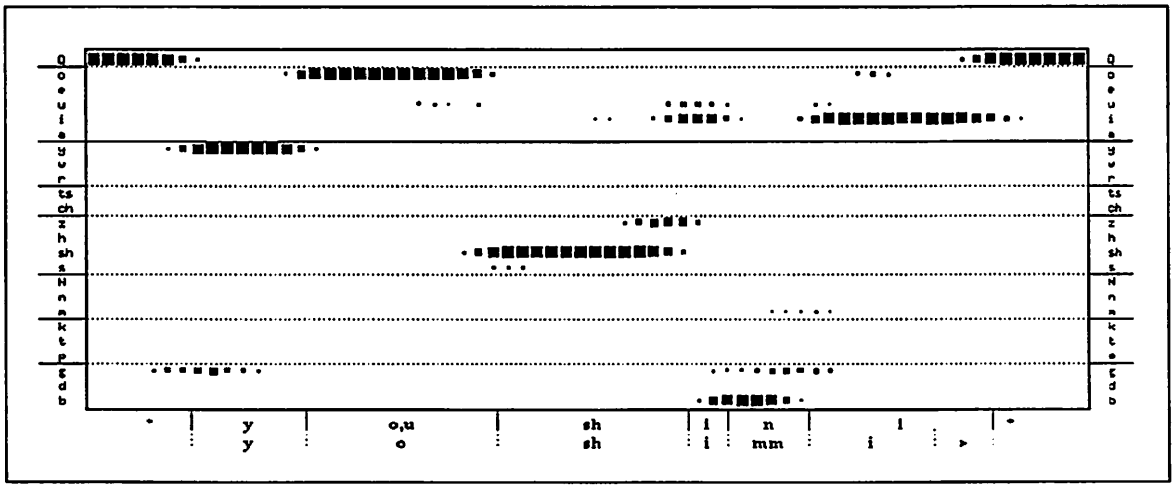

図 17: TDNN の出力パターン (youshini)

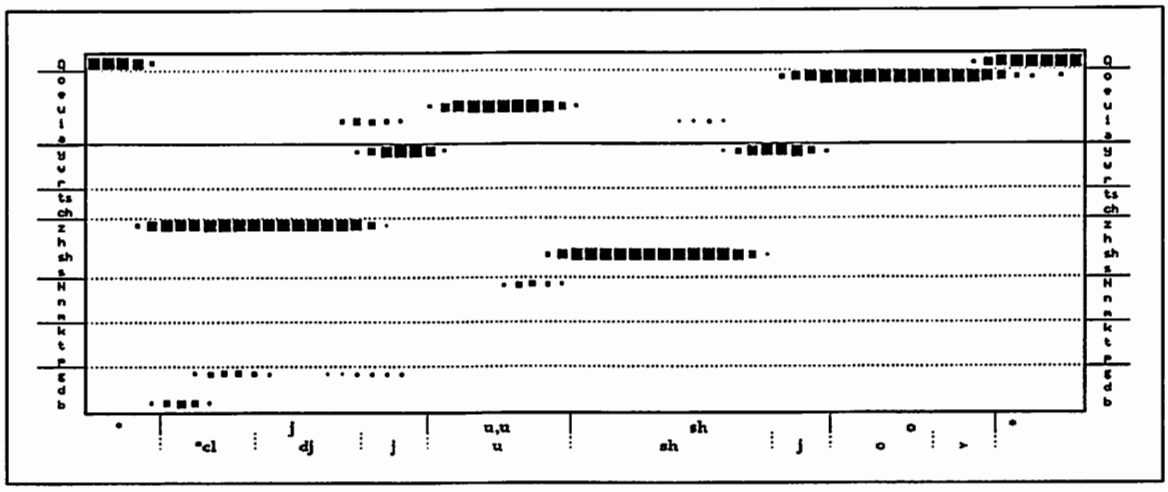

図 18: TDNN の出力パターン (juusho)

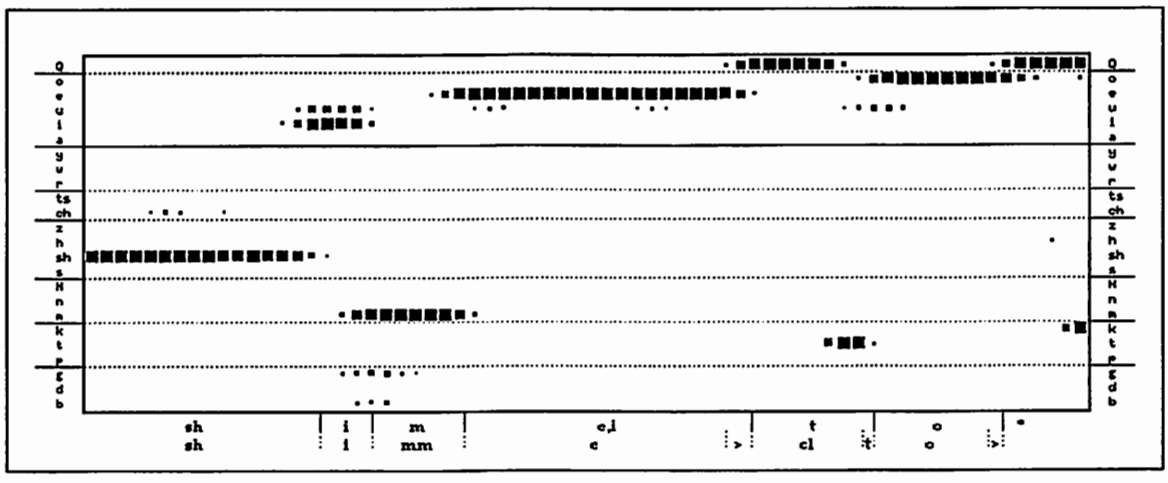

図 19: TDNN の出力パターン (shimeito)

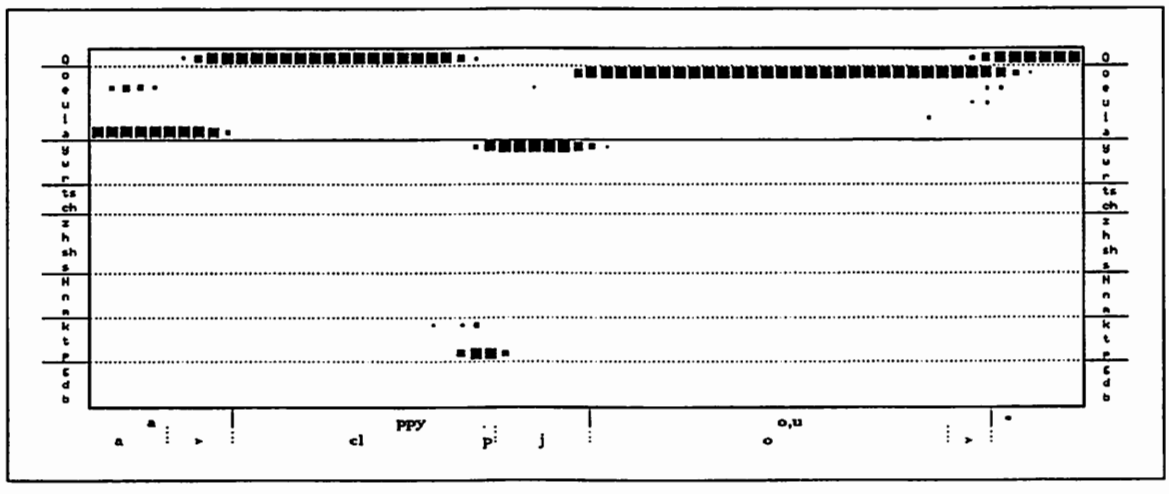

図 20: TDNN の出力パターン (happyou)

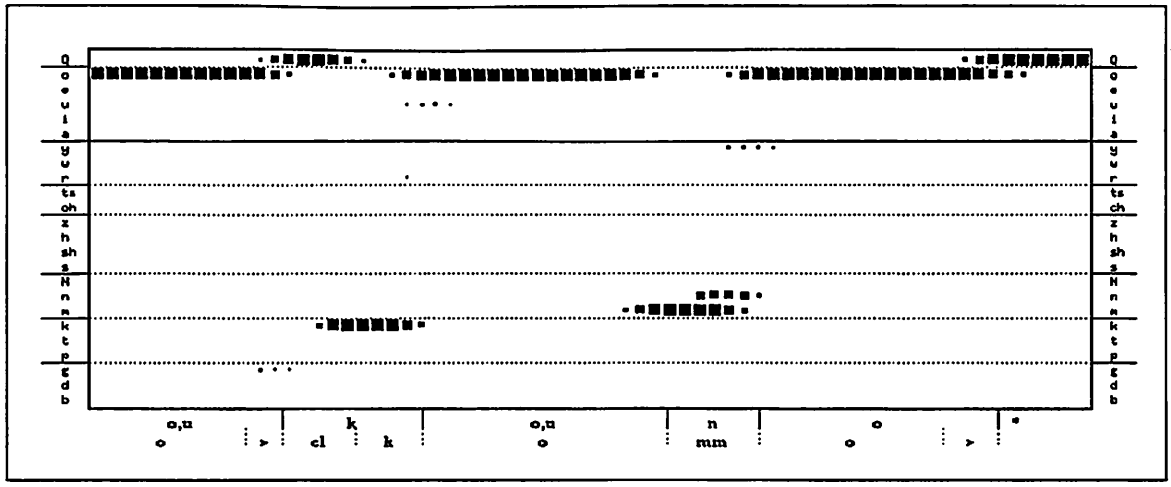

図 21: TDNN の出力パターン (choukouno)

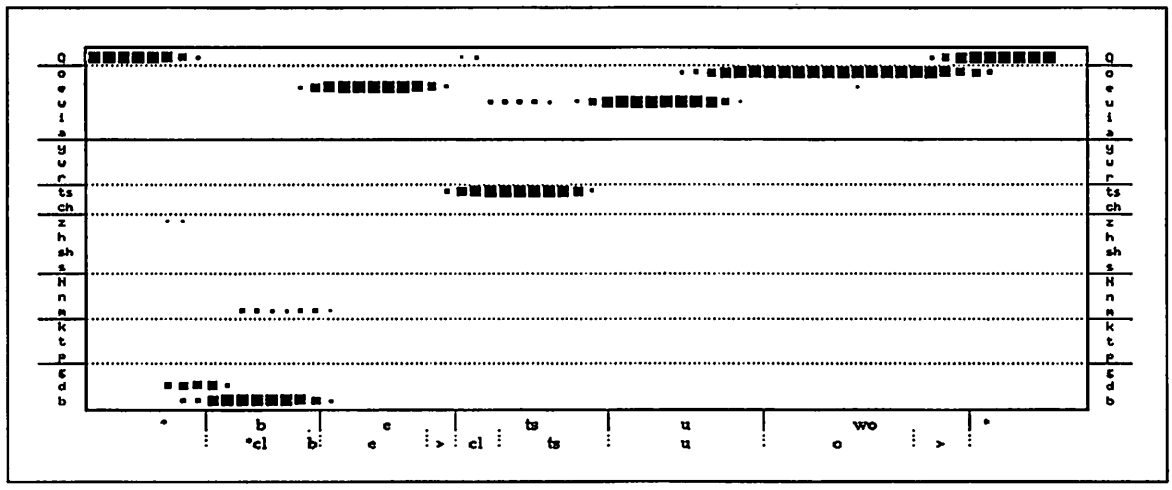

図 22: TDNN の出力パターン (betsuo)

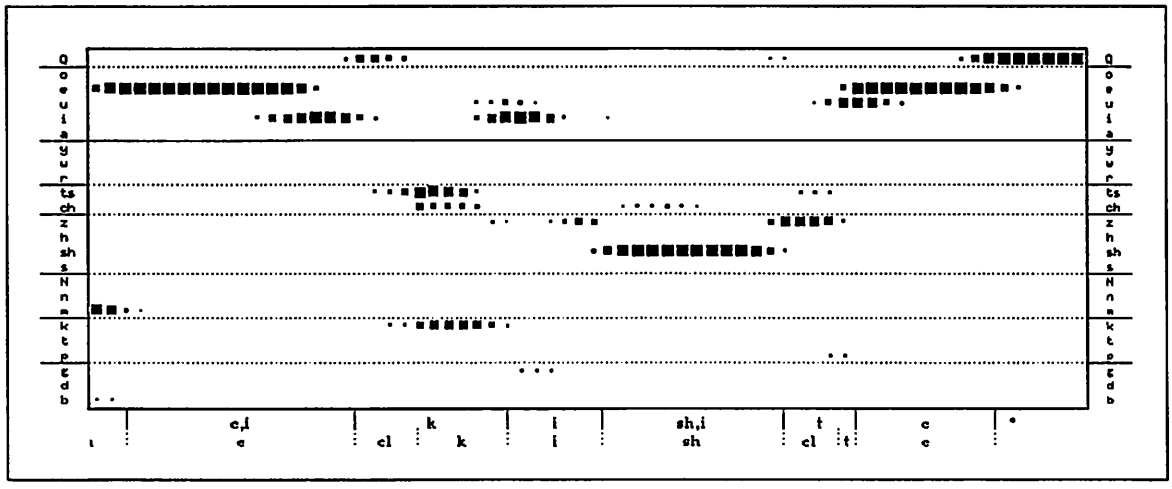

図 23: TDNN の出力パターン (meikishite)

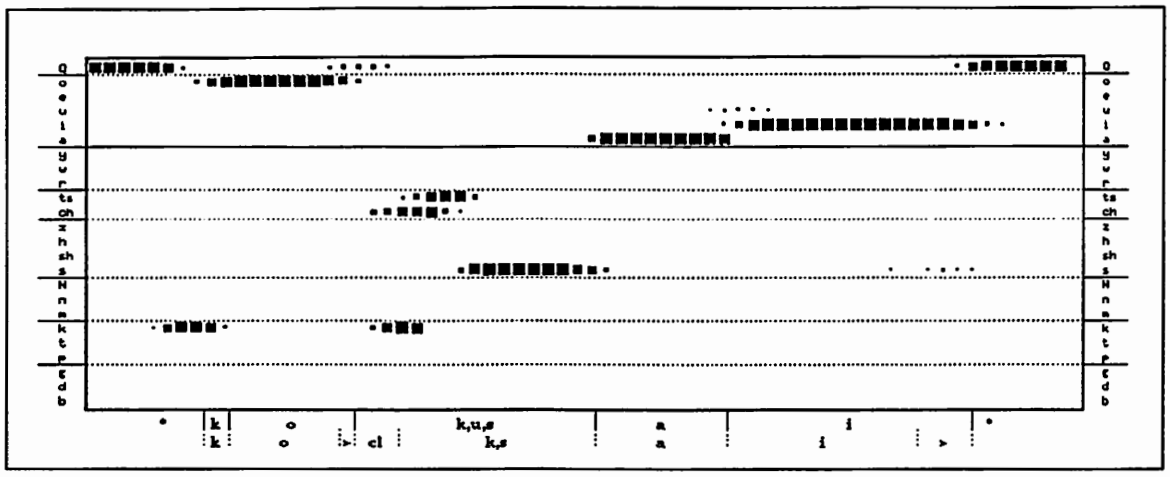

図 24: TDNN の出力パターン (kokusai)

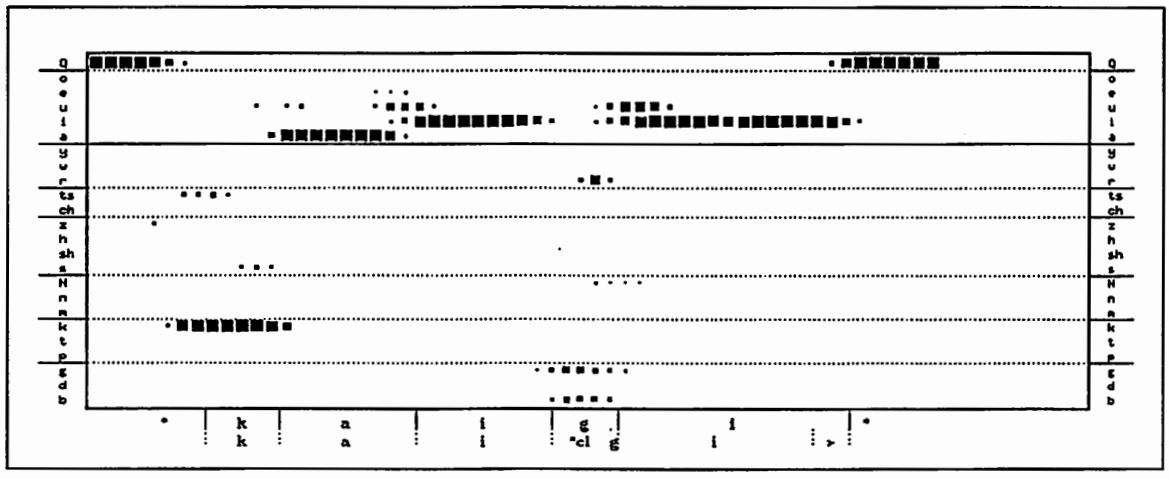

図 25: TDNN の出力パターン (kaigi)

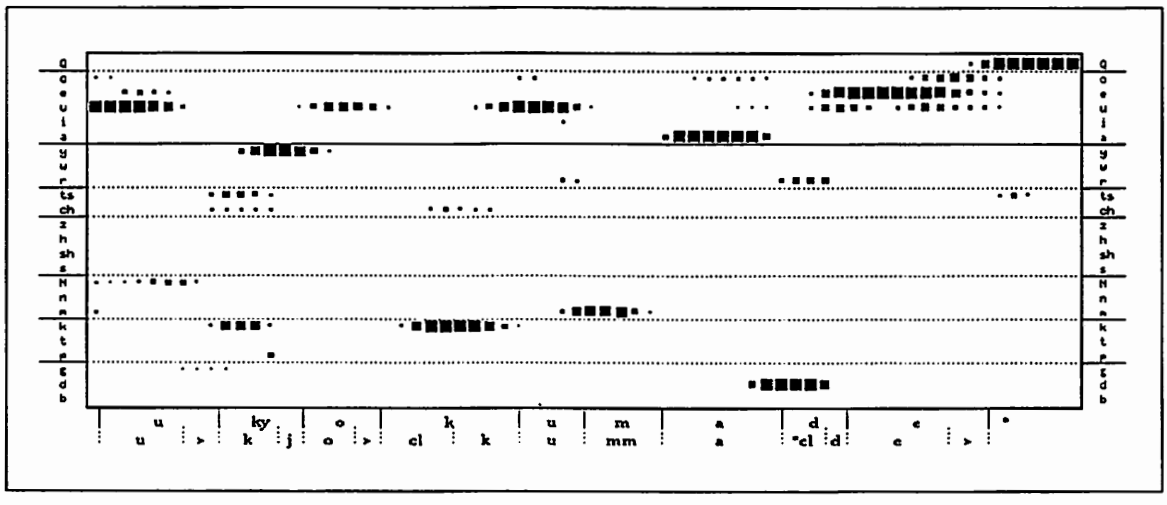

図 26: TDNN の出力パターン (jimukyokumade)

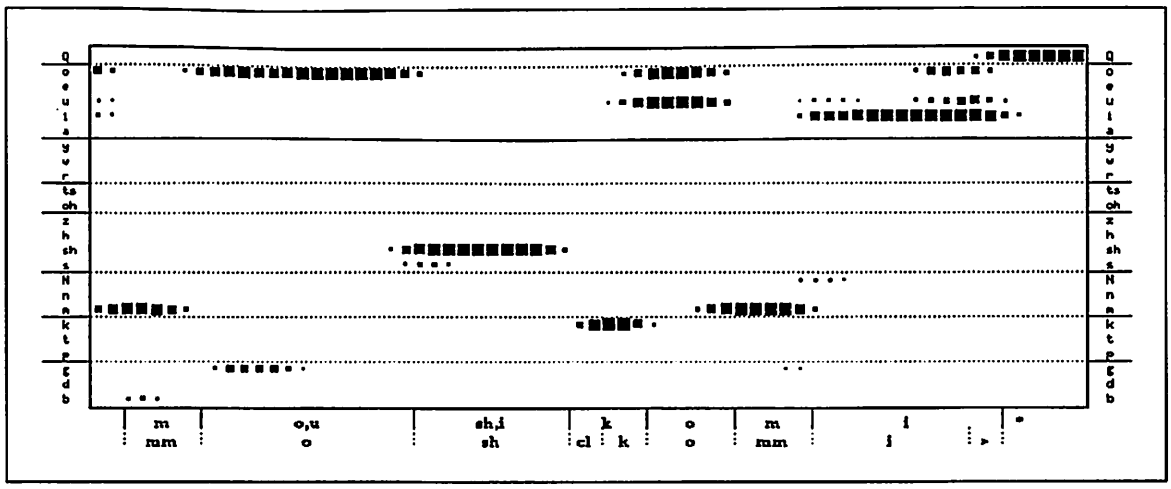

図 27: TDNN の出力パターン (omoushikomi)

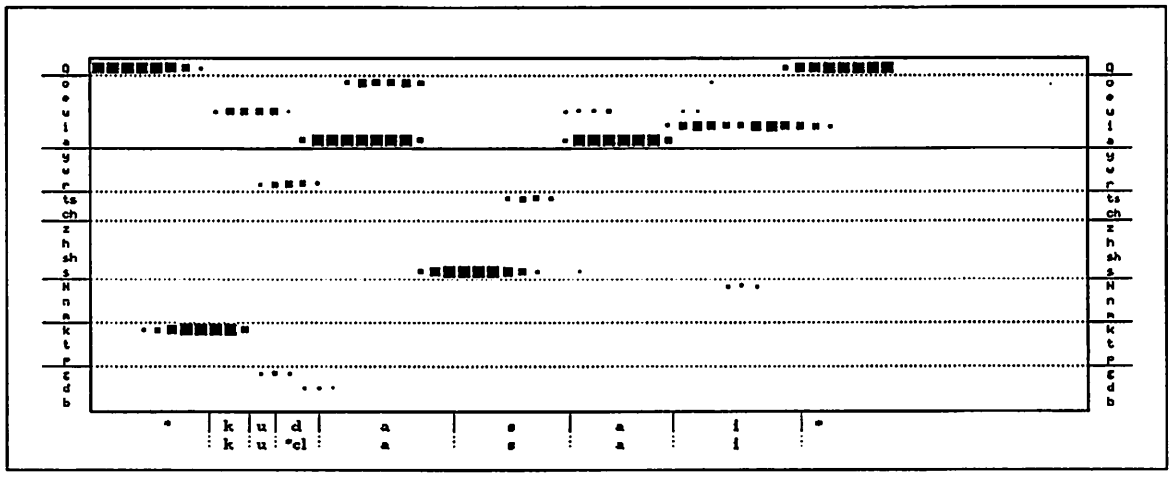

図 28: TDNN の出力パターン (kudasai)

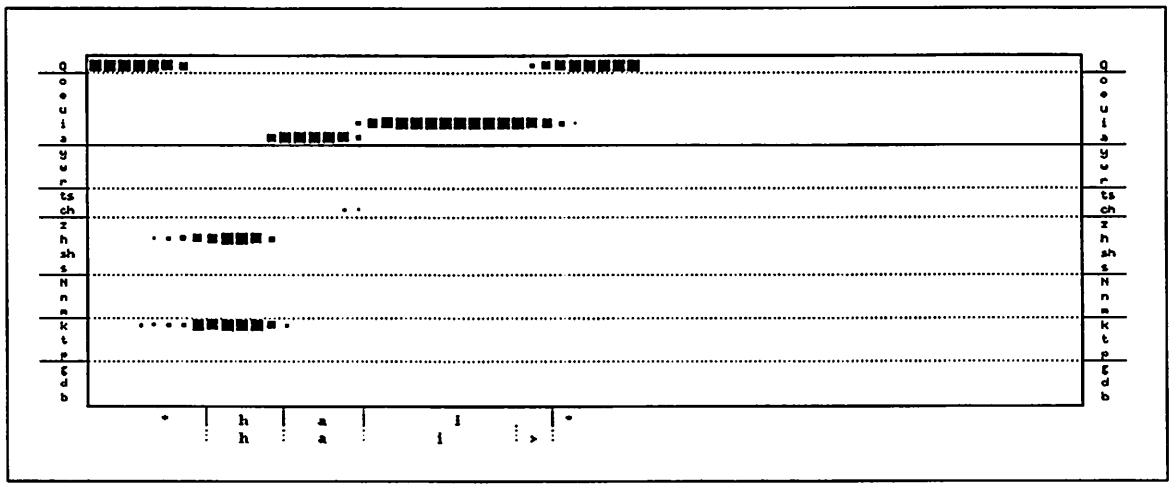

図 29: TDNN の出力パターン (hai)

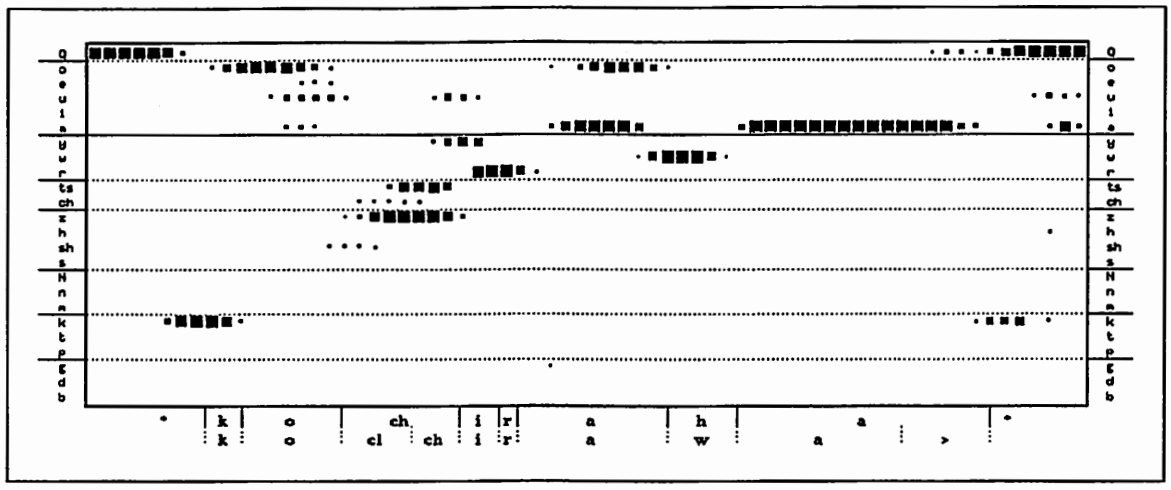

図 30: TDNN の出力パターン (kochirawa)

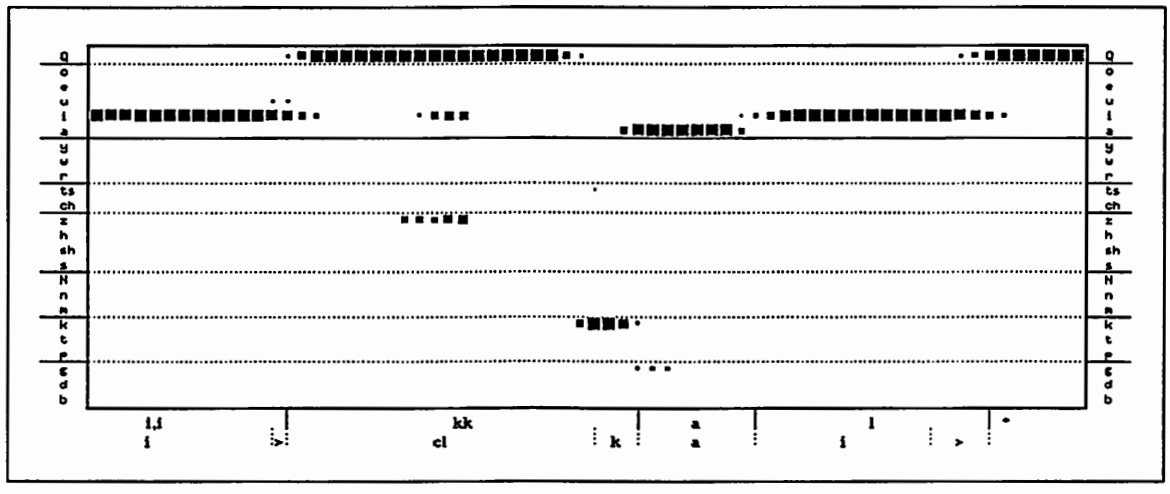

図 31: TDNN の出力パターン (daiikkai)

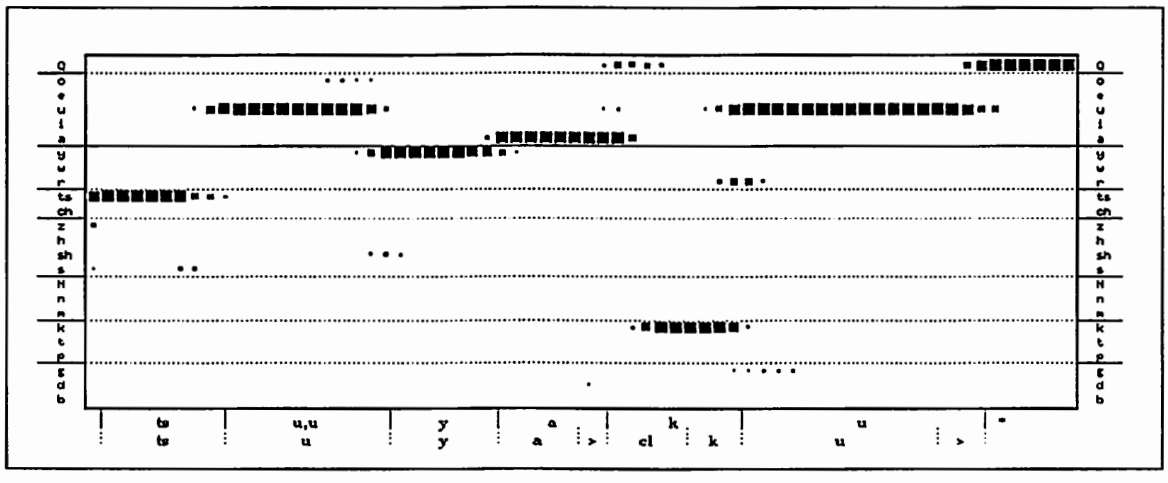

図 32: TDNN の出力パターン (tsuuyaku)

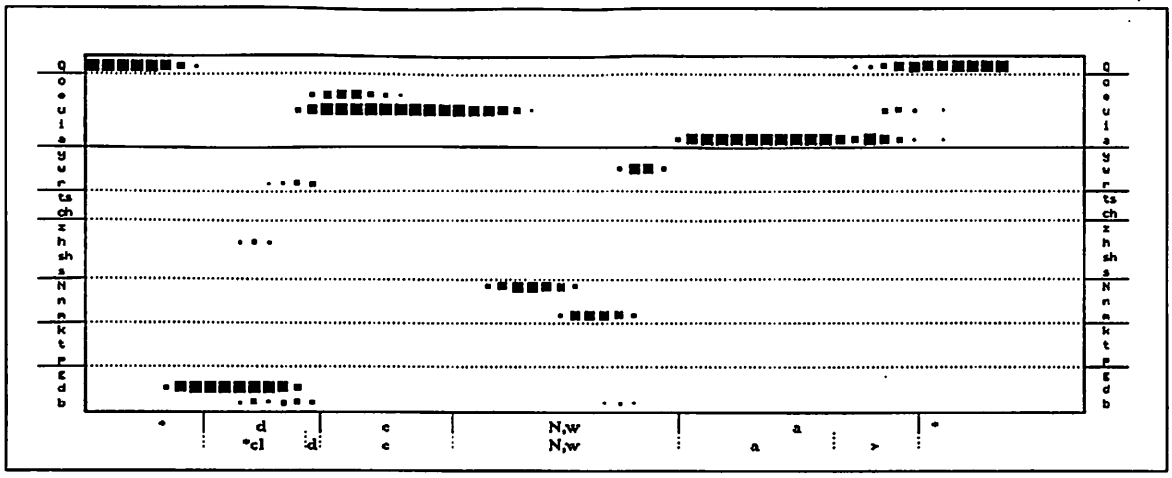

図 33: TDNN の出力パターン (deNwa)

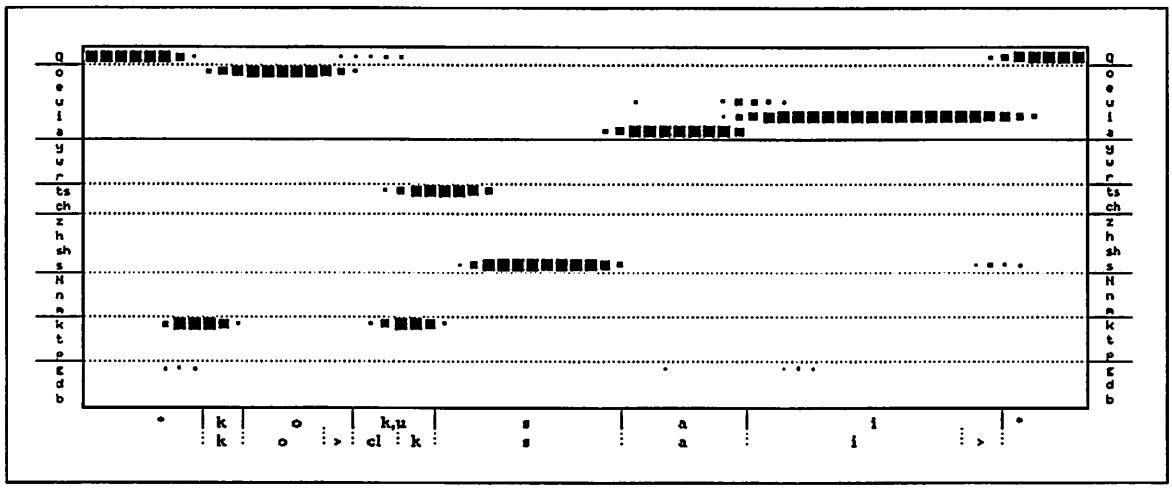

図 34: TDNN の出力パターン (kokusai)

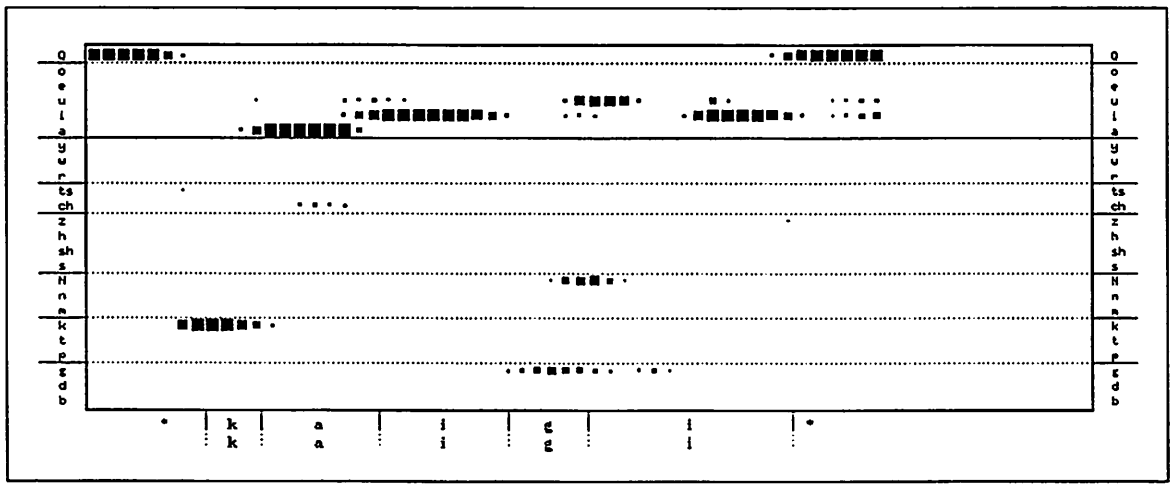

図 35: TDNN の出力パターン (kaigi)

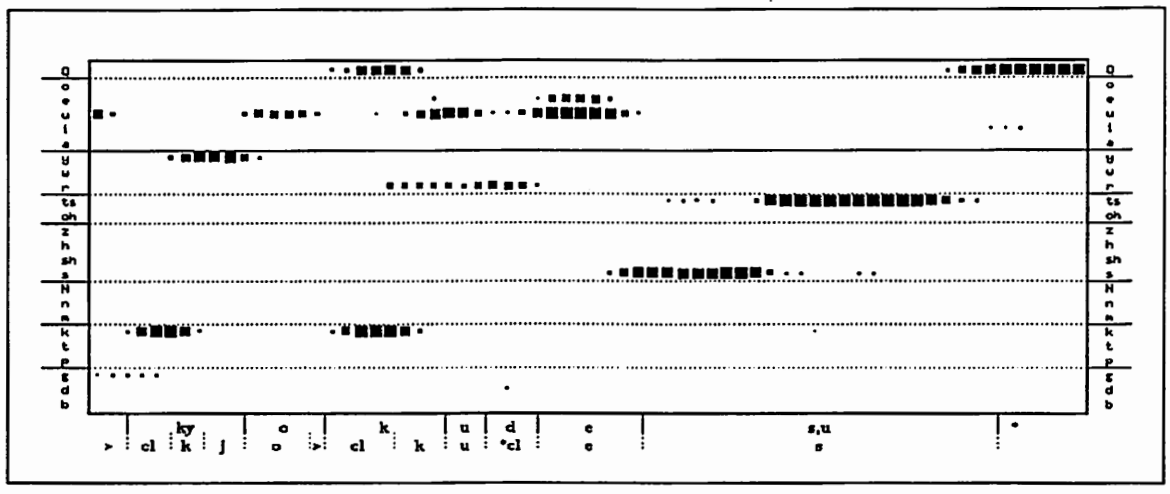

図 36: TDNN の出力パターン (jimukyokudesu)

B 実験に用いたソフトウエア

(TDNN + HMM を用いた文節音声認識)

```
(1) file.12k ------> file.FFT , file.OUT
cshell : sugi/tools/TDNN+LR/run.TDNN+LR

(2) file.OUT ------> file.FZY ( fuzzy VQ )
source : mickey/FVQ/src/vec_to_fzy.c
exec   : mickey/FVQ/src/vec_to_fzy
cshell : mickey/FVQ/exe.csh

(3) HMM + LR   音声認識
source : mickey/HMM+LR/src/parse.c
     LR.c
     ReadLR.c
     ReadRules.c
     adapt.c
     calidm.c
     cell.c
     dmviterbi.c
     error.c
     idur.c
     init_adapt.c
     io_mhmm.c
     logtbl.c
      scell.c
     temp.c
     time.c
     tophoneme.c

exec   : mickey/HMM+LR/src/Hmmlr_pmax
cshell : mickey/HMM+LR/rec_hmm+lr.all  etc.

(4) 認識率の計算
source : mickey/HMM+LR/Score/score.c
exec   : midkey/HMM+LR/Score/score

(5) duration を作る
source : (1) mickey/dur/src/dmviterbi.c
    calmveterbi.c
    io_mhmm1.c
    log_mhmm.c
 (2) mickey/dur/src/dchange.c
exec   : (1) mickey/dur/src/dmviterbi
 (2) mickey/dur/src/dchange
cshell : (1)
 (2) mickey/dur/exe.csh  etc.
```

(Alain Biem が実習で作成したプログラムの確認)

(6) label を作る
source : mickey/HMM/fvq/make_lab.c
exec : mickey/HMM/fvq/make_lab
cshell : Make_lab.all

(7) FVQ
source : ???
exec : mickey/HMM/fvq/Src/pcm_fzy_pmax
cshell : mickey/HMM/fvq/FZC.ALL
FZC.odd
FZC.trd

(8) training
source : ???
exec : mickey/HMM/hmm/Src/tfzmapf_pmax
cshell : mickey/HMM/hmm/train.all etc.

(9) recognition
source : ???
exec : mickey/HMM/hmm/rfzmapfn_k_pmax
cshell : mickey/HMM/hmm/rec.all etc.

(実験に用いたソフトウエアのまとめ)
(1) sugi/tools/TDNN+LR/???
(2) mickey/FVQ/src/vec_to_fzy
(3) mickey/HMM+LR/Src/Hmmlrfa_pmax
(4) mickey/HMM+LR/Score/score
(5) mickey/dur/src/dmviterbi
dchange
(6) mickey/HMM/fvq/make_lab
(7) mickey/HMM/fvq/pcm_fzy_wpd_pmax
(8) mickey/HMM/hmm/Src/tfzmapf_pmax
(9) mickey/HMM/hmm/rfzmapfn_k_pmax

(HMM + LR の認識結果)
/data10/biem/rec_result/new_hmm_dur にある。