TR-IT-0044

時間フレームの依存性を考慮したHMM

西村孝則
Takanori NISHIMURA

Helmut Lucke

1994.2

現在、 HMM(Hidden Markov Model) を用いた音声認識では、主に時間フレーム間は独立として扱っている。しかし、時間フレーム間の依存性があるのではないかと思われる。この、時間フレーム間の依存性に関する研究は、まだ、あまりなされていない。本稿では、時間フレーム間の依存性を明確にするため行なった研究の結果を報告する。この研究は、西村が豊橋技術科学大学の実習生としてＡＴＲに滞在中に行なったものである。

本報告では、始めに、現在までに行なわれている音声認識の概要を述べる。次に、現在の HMM の問題点について述べ、その解決への可能性について検討する。また、本研究で提案した方法について、音声認識の実験を行なった。この結果をもとに従来法と本方式の認識率を比較をする。

目次

1 本研究の音声認識の概要

1.1 音声認識のための前処理

音声はアナログ信号である。しかし、コンピュータが扱えるのは、ディジタル信号のみである。したがって、音声認識に入る前に前処理が必要である。また、アナログ信号をディジタル信号に変換しただけでは、音声認識において適切な情報とは限らない。音声認識しやすいデータに変換する前処理が必要となる。以下に、本研究で使用されている前処理について述べる。

(1.1.1) A/D 変換

アナログ信号をディジタル信号に変換するには、その対応づけが必要である。アナログ信号は時間軸方向、振幅方向ともに連続である。しかし、デジタル信号にするには、その両方とも離散化する必要がある。時間軸方向の離散化をサンプリングと言い、振幅方向の離散化を量子化と言う。

本研究に使われている音声データは、サンプリング周波数 16kHz、量子化ビット数 12bits である。

(1.1.2) 特徴抽出変換

A/D 変換によって、得られた情報は、ただ単に信号の振幅の時間系列である。しかし、音声認識においては、もっと音声認識に適した情報に変換することが多い。すなわち、特徴抽出した音声データを用いて音声認識することが多い。本研究では、その変換法に LPC メルケプストラム変換を用いている。

この変換において、音声信号は 256bytes づつ取り出される。その 256 個の振幅情報から、特徴抽出された結果として、16 次元のベクタを得る。このとき、特徴抽出は 10ms ごとに行なう。したがて、特徴抽出は少しづつ重なりあった音声信号を用いることになる。音声認識は、この結果を用いて行なう。以後、音声データといえば断らない限りこの特徴抽出されたデータのことを言う。

1.2 音声学習

本研究では、音声モデルとして HMM(Hidden Markov Model) を用いる。このモデルを用いるためには、その 隠れマルコフ網 (Hidden Markov network,HMnet) [1] が必要となる。この HMnet には、状態遷移確率やシンボル放出確率などが付追している。この HMnet の状態遷移形態や確率は、始めから決まっているものではなく、扱う環境 (言語や話者など) で大きく異なる。従って、環境に合わせて最適設定する必要がある。以下に、音声学習に関することについて簡単に述べる。

(1.2.1) HMnet の生成

前に述べたように HMM を用いるためには、その状態遷移を示す HMnet が必要である。HMnet の生成法には、

1. 音声データによらずどの音声に対しても同じ network となるように機械的に生成する方法

2. 今までの結果と感をたよりに、最適だと思われるように手動で生成する方法

3. 音声データを元に、ネットワーク全体の遷移確率が最大になるように自動生成する方法

などが挙げられる。1 の方法では、音声データ (文脈) に依存しない HMnet となる。それ以外の方法では、ある程度、音声データに依存した network になる。今回の実験では、1 の方法を用いた。

(1.2.2)　HMnet のパラメータの設定

HMnet のパラメータには、状態遷移確率、シンボル放出確率、期待値、分散などがある。これらのパラメータの決定は、音声データを元にその音声データと正解の音素を network に入力した時の全体の確率が、最大になるように決定する。この、network のパラメータ決定のことを音声学習と言う。HMM においては、各パラメータは、環境によって決まるものであるので、この音声学習は必ず必要となる。

1.3　音声認識

音声学習を済ませた HMnet は、そこで始めて認識に使用することができる。いままでは、いわば音声認識するための前処理である。

音声認識の方法は、音声データを HMnet に入力した時に、 network 全体を通した確率が最大となる path を求めることにより実現する。その時に HMnet が出力したシンボルが認識結果となる。すなわち、以下の式を満たす時の出力シンボルが認識結果となる。

$$P(\boldsymbol{O}) = \max_{\boldsymbol{S}} \prod_{t=1}^{N} P(S_t|S_{t-1}) \cdot P(O_t|S_t) \tag{1}$$

但し、　$\boldsymbol{O} = O_1, \ldots, O_N$は観測シンボル系列
$\boldsymbol{S} = S_1, \ldots, S_N$は状態遷移系列
Nは系列の長さ

この network 全体の確率が最大となる path を求める方法には、Viterbi アルゴリズム [2] などを用いる。

ここで、音声認識と言ってもただ単に音素だけを認識することだけにこだわらなくてもよい。言語によっては、あり得ない発音などがある。したがって、音声認識過程において、そのような組合せになるものは、無視するようにすれば認識結果が向上することが予測される。

このような方法を用いることにより、音声認識を行なう。

2　現在用いられている HMM の問題点

現在、HMM で扱われている確率は以下のようである [3]。

$$P = P(S_0)P(O_0|S_0) \prod_{t=1}^{N} P(S_t|S_{t-1})P(O_t|S_t) \tag{2}$$

この式の状態の依存関係をグラフで表すと図 1のようになる。

この図を見てわかるように現在までは、前の状態 S_{t-1} のみで次の状態 S_t と 観測シンボル O_t を決定している。よって、O_{t-1} と O_t は独立として扱っている。しかし、実際には O_{t-1} と O_t の依存関係もあるのではないかと考える。この O_{t-1} と O_t を独立として扱うことでの問題点は

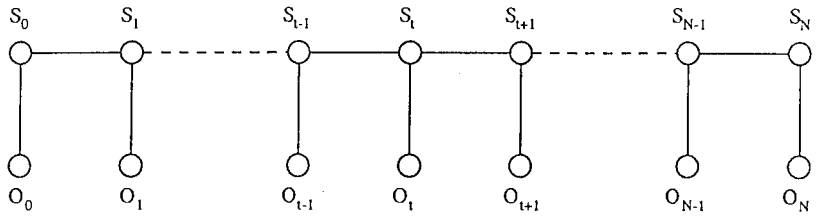

図 1: 従来法の依存グラフ

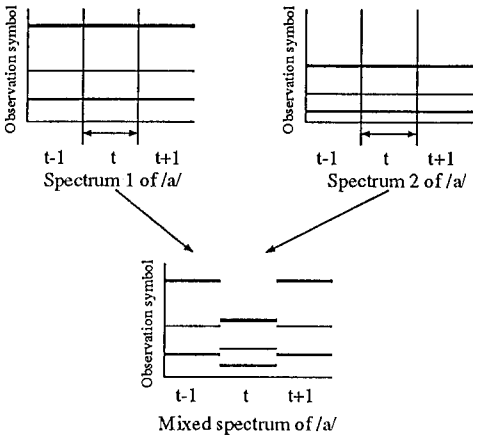

図 2: Phoneme のスペクトルの違いによる影響

1. Phoneme のスペクトルの違いによる影響

2. Phoneme の長さの違いによる影響

があげられる。

例を挙げると、前者は、図2 のように、同じ音素 /a/ でもスペクトルが異なることがある。この場合、時間フレームを独立として扱うと合成したスペクトルでも認識してしまう。この合成したスペクトルは実際にはあり得ない。よって、これが誤認識の原因となるのではないかと考える。

後者は、図3 のように、同じ音素 /a/ でも図(a) のように短い場合もあれば、図(b) のように長い場合もある。この場合の確率を計算すると、次のようになる。

$$\begin{aligned} \text{(a)}\ P_{/\text{a}/-(\text{a})} &= \{P(S_t|S_{t-1})P(O_t|S_t)\}^n \\ \text{(b)}\ P_{/\text{a}/-(\text{b})} &= \{P(S_t|S_{t-1})P(O_t|S_t)\}^{2n} \end{aligned}$$

この式からわかるように、同じ音素 /a/ であっても、短い音素の時は確率が高く、長い音素の時は確率が小さくなる。この違いは、長さに対して、指数的に反比例し、確率が大きく異なる。

これらは、従来の HMM は、O_{t-1} と O_t の間に何ら関係を持たせていない。すなわち、O_{t-1} と O_t を独立として扱っているので起きる問題だと考える。したがって、モデル化において O_{t-1} と O_t の間に関係を持たせるようにすると認識率が上がるのではないかと考える。

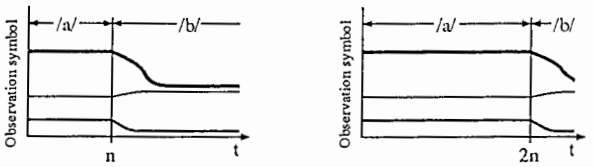

図 3: Phoneme の長さの違いによる影響

3　問題点解決への可能性

前節で述べた問題点

O_t と O_{t-1} を独立として扱っている点

につて解決する方法は、現在までにいくつか研究はなされている。いかにそれらの方法を挙げる。そして、本方式の方法について述べる。

3.1　Wellekens の方法

Wellekens は改善する方法として (2) 式を以下のように書き換えた [4]。

$$P = P(S_0)P(O_0|S_0)\prod_{t=1}^{N} P(S_t|S_{t-1})P(O_t|O_{t-1}, S_{t-1}, S_t) \tag{3}$$

この式を書き換えると、

$$\begin{aligned} P &= P(S_0)P(O_0|S_0)\prod_{t=1}^{N} P(S_t|S_{t-1})P(O_t|O_{t-1}, S_{t-1}, S_t) \\ &= P(S_0)P(O_0|S_0)\prod_{t=1}^{N} P(S_t|S_{t-1})\frac{P(O_t, O_{t-1}|S_{t-1}, S_t)}{P(O_{t-1}|S_{t-1})} \\ &= P(S_0)P(O_0|S_0)\prod_{t=1}^{N} \frac{P(O_t, O_{t-1}, S_t|S_{t-1})}{P(O_{t-1}|S_{t-1})} \\ &= P(S_0)P(O_0|S_0)\prod_{t=1}^{N} P(O_t, S_t|O_{t-1}, S_{t-1}) \end{aligned}$$

よって、この方法の依存関係はグラフで表すと図 4 のようになる。この方法では、$P(O_t|O_{t-1}, S_{t-1}, S_t)$ の条件付確率を求める時のことを考えると、変数が O_t,O_{t-1},S_{t-1},S_t と 4 個もあり、パラメータの数が増えるので、パラメータの推定が難しい。

3.2　TAKAHASHI 等の方法

TAKAHASHI 等は改善する方法として (2) 式を以下のように書き換えた [5]。

$$P = P(S_0)P(O_0|S_0)\prod_{t=1}^{N} \frac{P(S_t|S_{t-1})P(O_t|S_t)P(O_t|O_{t-1})}{\alpha} \tag{4}$$

この方法の依存関係はグラフで表すと図 5 のようになる。この方法では、$\sum P = 1$ になるようにするために、正規化係数 α を導入している。

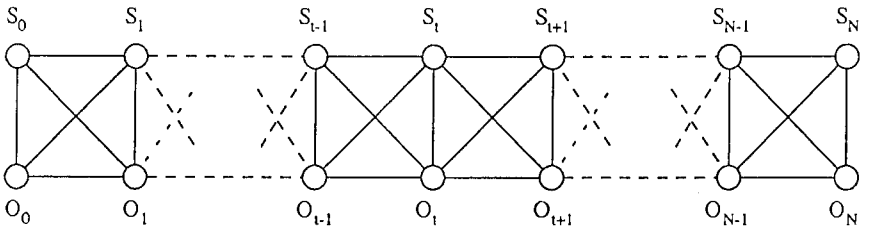

図 4: Wellekens の方法の依存グラフ

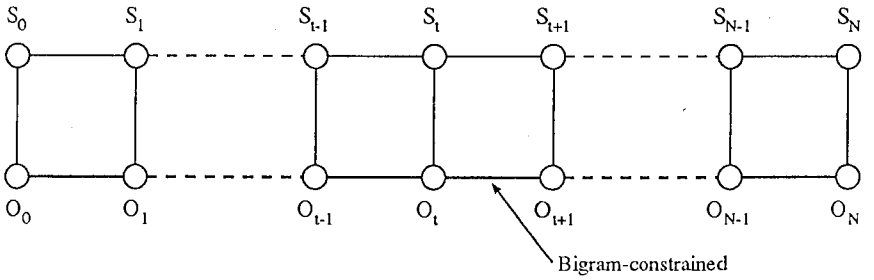

図 5: TAKAHASHI 等の方法の依存グラフ

3.3 本研究の方法

本研究では、O_t と O_{t-1} に依存関係を持たせるために $P(O_t|S_t)$ を $P(O_t|O_{t-1}, S_t)$ に置き換えた。すなわち、

$$P = P(S_0)P(O_0|S_0)\prod_{t=1}^{N} P(S_t|S_{t-1})P(O_t|O_{t-1}, S_t) \tag{5}$$

とした。この方式の依存関係はグラフで表すと、図 6 のようになる。

この方法を先に述べた方法と比較すると、条件付確率を求めるとき、最大で 3 個の変数を扱えばよい。また、正規化係数 α を必要としない点などが利点となる。

3.4 従来のプログラムへの適応

従来のプログラムに適用するための工夫について述べる。

(2) 式において入力として与えられる $[O_t]$ を $[O_{t-1}O_t]$ とする。この操作により、(2) 式は

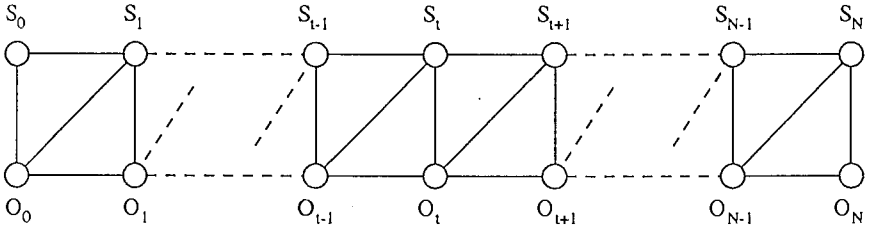

図 6: 本方式の依存グラフ

次のようになる。

$$P' = P(S_0)P(O_{-1}, O_0|S_0)\prod_{t=1}^{N} P(S_t|S_{t-1})P(O_{t-1}, O_t|S_t) \tag{6}$$

また、$P(O_{t-1}, O_t|S_t)$ から $P(O_{t-1}|S_t)$ は、

$$P(O_{t-1}|S_t) = \int_{-\infty}^{\infty} P(O_{t-1}, O_t|S_t)dO_t \tag{7}$$

より求めることができる。このようにすることにより、(5) 式の $P(O_t|O_{t-1}, S_t)$ は、

$$P(O_t|O_{t-1}, S_t) = \frac{P(O_{t-1}, O_t|S_t)}{P(O_{t-1}|S_t)} \tag{8}$$

より、求めることができる。

このように、入力を $[O_t]$ から $[O_{t-1}O_t]$ にし、$P(O_t|O_{t-1}, S_t)$ を (8) 式のように計算するようにすれば従来のプログラムでも、次の確率が最大となる処理を行なうようになる。

$$P_{\text{adapt}} = P(S_0)P(O_0|O_{-1}, S_0)\prod_{t=1}^{N} P(S_t|S_{t-1})P(O_t|O_{t-1}, S_t) \tag{9}$$

ここで、(5) 式の $P(O_0|S_0)$ が (9) 式では $P(O_0|O_{-1}, S_0)$ となっている。しかし、これは条件の部分に O_{-1} があるだけなので、$N \gg 1$ であれば、全体に対する影響は少ないと考える。

4　実験方法

4.1　HMnet

本研究では、HMM を実現する方法として、隠れマルコフ網 (Hidden Markov Network: HMnet) を用いた。HMnet は、本研究所で開発した SSS-ToolKit が扱うことのできる形式のものを用いた [6]。

4.2　本研究で扱う HMnet

本研究で扱う HMnet は、

> 音声データによらずどの音声に対しても同じ network となるように機械的に生成する方法

である。これは、どの音素に対しても、状態遷移数、各状態の分布の混合数が同じであることを示す。たとえば、状態遷移数を 3、混合数を 5 と決めた場合、HMnet は、図 7のようになる。本研究では、このような HMnet を用いて実験を行なった。

4.3　実験方法

実験は、時間フレームの依存性を考えない場合 (frame independent) と、依存性を考えた場合 (frame dependent) の両者について行なった。

HMnet としては、次のようにした。

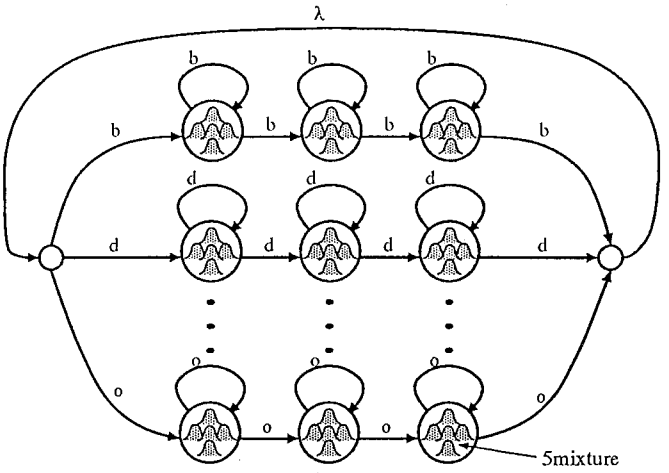

図 7: HMnet の例 (状態遷移数 3, 混合数 5)

音素 (Phoneme) : b d g p t k m n ng s sh h z ch ts zh r w j a i u e o q –
状態数 (State Number) : 3
混合数 (Mixture Number): 1,3,5,7,10,15

音声学習は、この HMnet を用いて、以下の 3 つの条件で行なった。

1. Speaker Dependent(音声学習が 1 人の場合)

2. Multiple Speaker Dependent(音声学習が 2 人の場合)

3. Multiple Speaker Dependent(音声学習が 8 人の場合)

音声学習の入力音声データは、音声データの偶数番目の単語を用いた。学習に使用した単語数は、学習音声が 1 人の場合と 2 人の場合は、偶数番目の全ての単語 (1 人当たり 2620 単語)、音声学習が 8 人の場合は、偶数番目の単語でランダムに 400 単語 (ただし、26 種類全ての Phoneme を含んでいる) を使用した。まとめると、以下の表のようになる。

学習人数 Number of training speakers	一人当たりの学習単語数 Number of Trianing words by a speaker	学習全体での学習単語数 Number of training words
1	2620	2620
2	2620	5240
8	400	3200

音声認識は、音素の認識と単語の認識を行なった。音素認識は、音声データに音素の区切りの情報を付加したもので行なった。単語認識は、音声データには区切りの情報を入れず連続音声データとし、音素 FSA(Phonemic Final State Automaton) で表現される音素連結規則のみを用いて行なった。また、単語辞書などは使用しないで単語認識を行なった。

音声認識では、以下の音素の認識を行なった。

- b d g p t k m n ng s sh h z ch ts zh r w j a i u e o

音声学習と違うのは、q 音素と－(silence) 音素がないことである。認識音素は、音声データの奇数番目の単語を音素認識に使用した。また、認識音素数としては、1 音素当たり原則 100 音素とした。ただし、p 音素と w 音素は、音声データの数が足りなかったため、それぞれ 28 音素、89 音素とした。

認識の音素	認識に使用した音素数
p	28
w	89
p,w 以外	100

単語認識では、音声データの奇数番目の単語を使用し、その中から、ランダムに 2000 単語あるいは 200 単語を単語認識に使用した。

5　実験結果

5.1　Speaker Dependent(音声学習が 1 人の場合)

音声学習に使用した音声データは、“MTK”,“FKN” である。音声認識は、“MTK” の音声データを使用して学習した HMnet と “FKN” の音声データを使用して学習した HMnet の双方で、“MTK” と “FKN” の認識を行なった。

音声学習	音声認識	
“MTK”	“MTK”	“FKN”
“FKN”	“MTK”	“FKN”

音素認識の結果を図 8 と図 9 に示す。

また、単語認識の結果を図 10 と図 11 に示す。単語認識での insertion 誤り、deletion 誤り、substitution 誤りの関係を図 12、図 13、図 14、図 15、図 16、図 17 に示す。

5.2　Multiple Speaker Dependent(音声学習が 2 人の場合)

音声学習に使用した音声データは、“MTK”,“FKN” である。音声認識は、“MTK” と “FKN” の音声データを使用して学習した HMnet で “MTK” と “FKN” の認識を行なった。

音声学習	音声認識	
“MTK” and “FKN”	“MTK”	“FKN”

音素認識の結果を図 18 に示す。

また、単語認識の結果を図 19 に示す。単語認識での insertion 誤り、deletion 誤り、substitution 誤りの関係を図 20、図 21、図 22 に示す。

5.3　Multiple Speaker Dependent(音声学習が 8 人の場合)

音声学習に使用した音声データは、

“MTK”,“FKN”,“FMS”,“FTK”,“FYM”,“MAU”,“MHT”,“MXM”

である。音声認識は、これら 8 人の音声データを使用して学習した HMnet で幾人かの音声認識を行なった。認識に使用した音声データはつぎのようである。

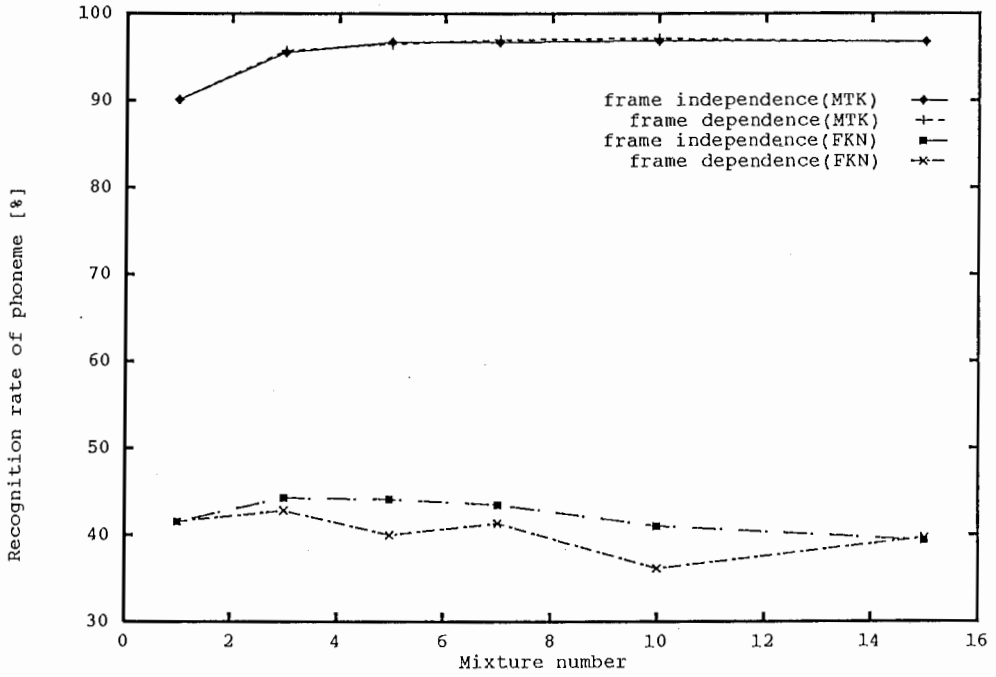

図 8: MTK による学習の音素認識結果

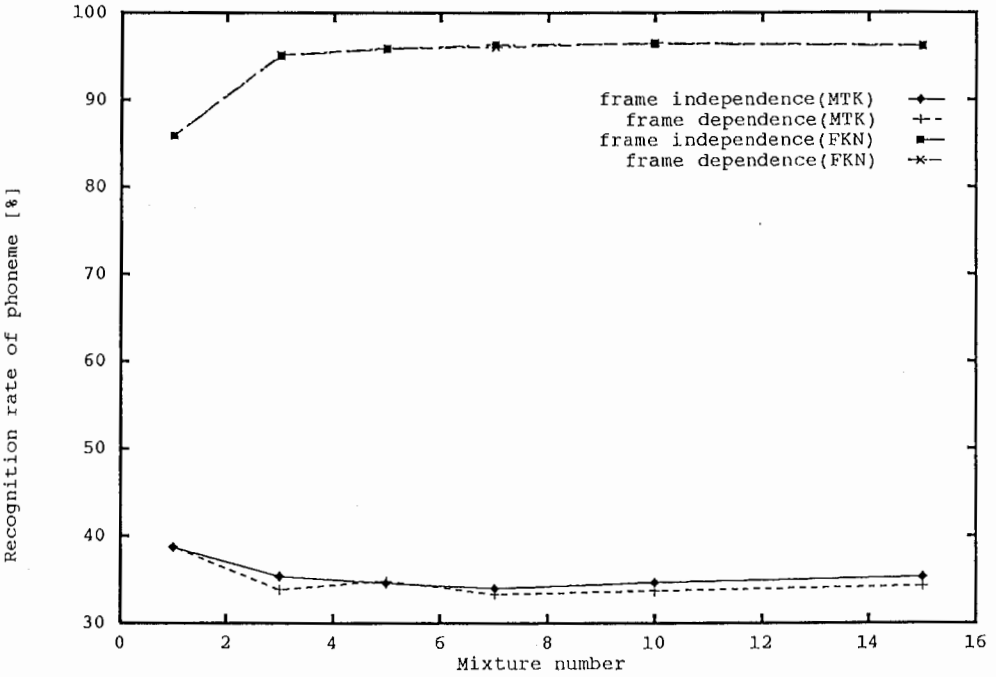

図 9: FKN による学習の音素認識結果

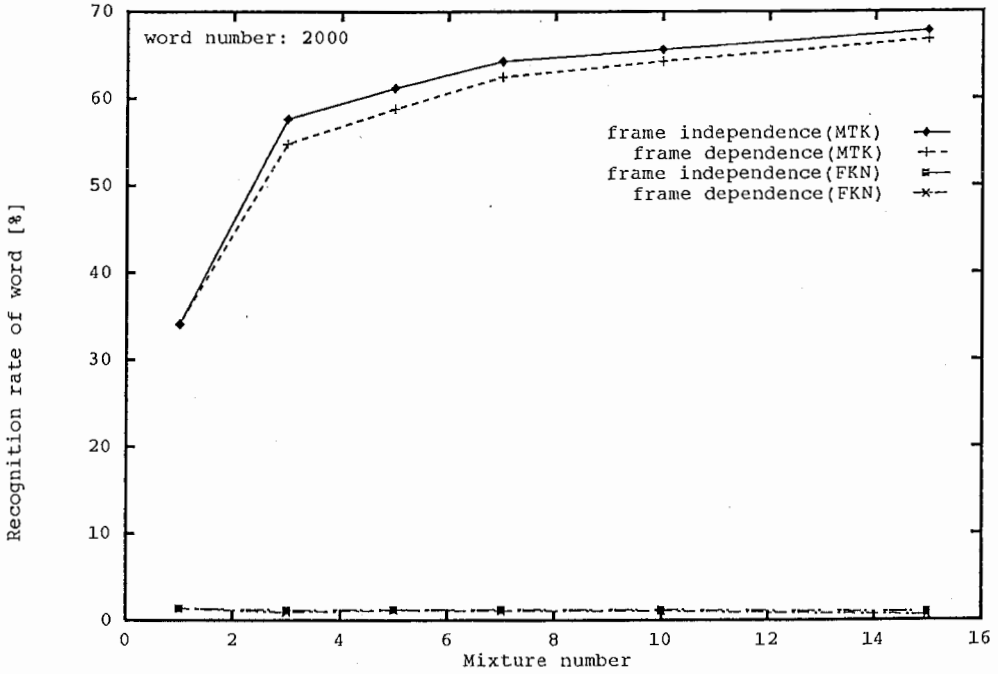

図 10: MTK による学習の単語認識結果

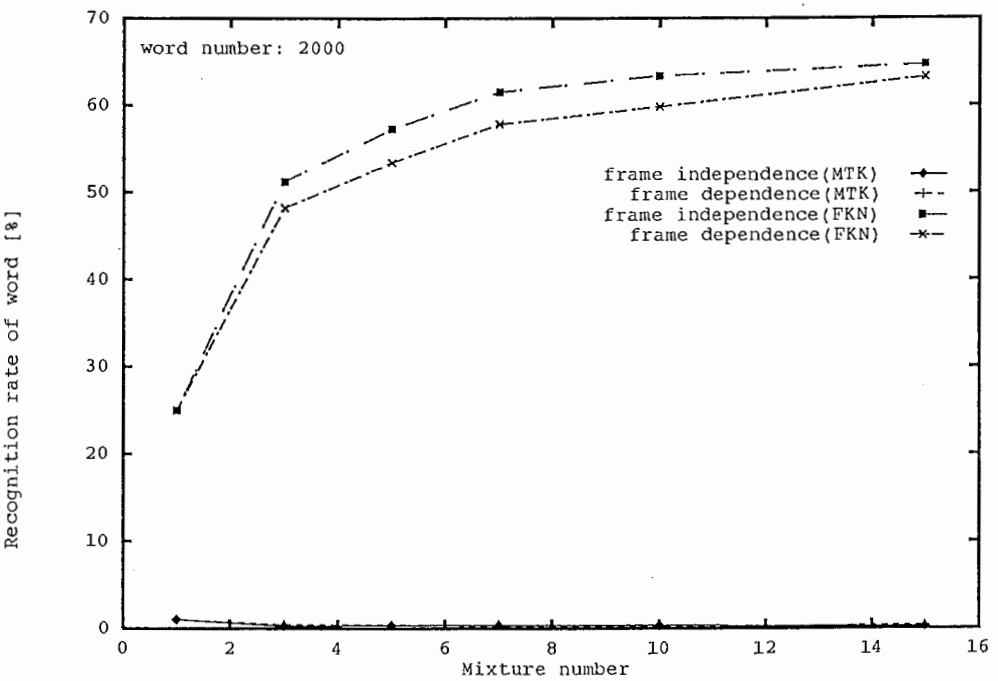

図 11: FKN による学習の単語認識結果

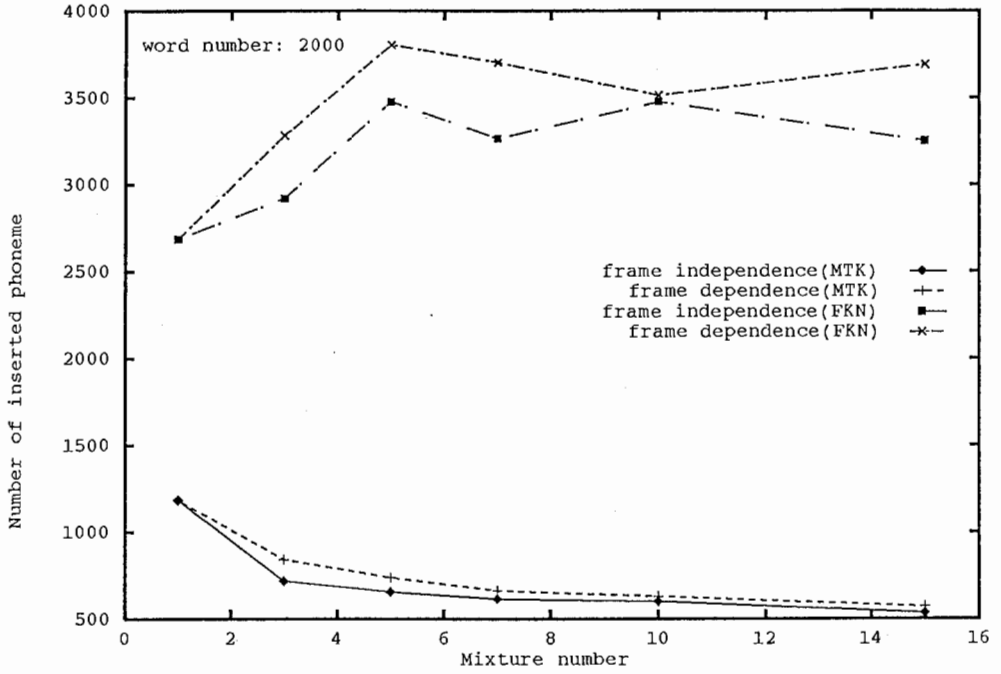

図 12: MTK による学習の単語認識での insertion 誤り結果

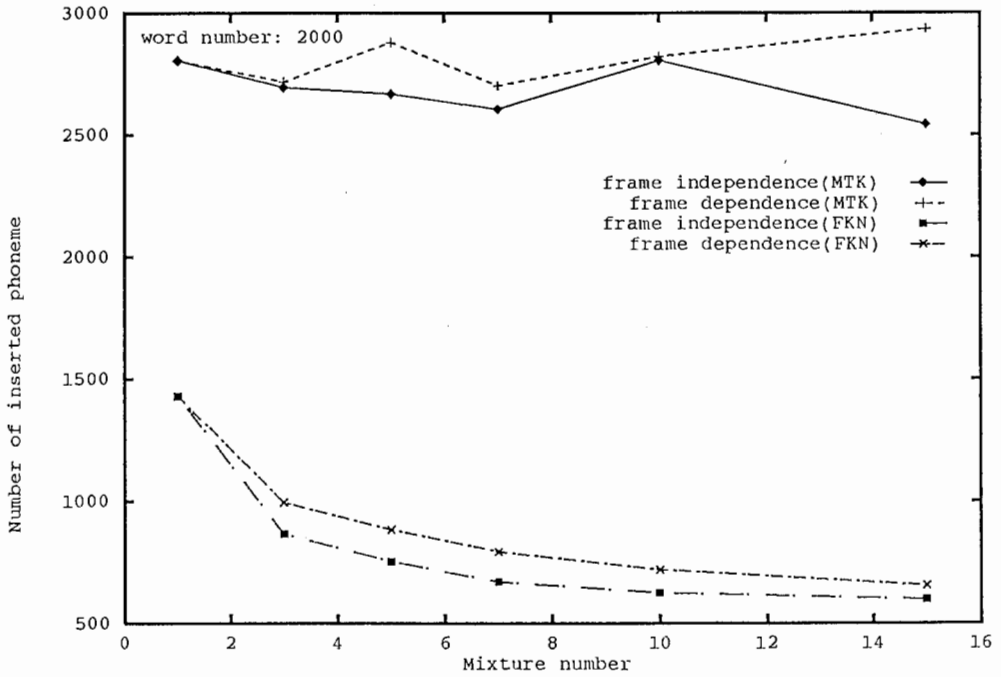

図 13: FKN による学習の単語認識での insertion 誤り結果

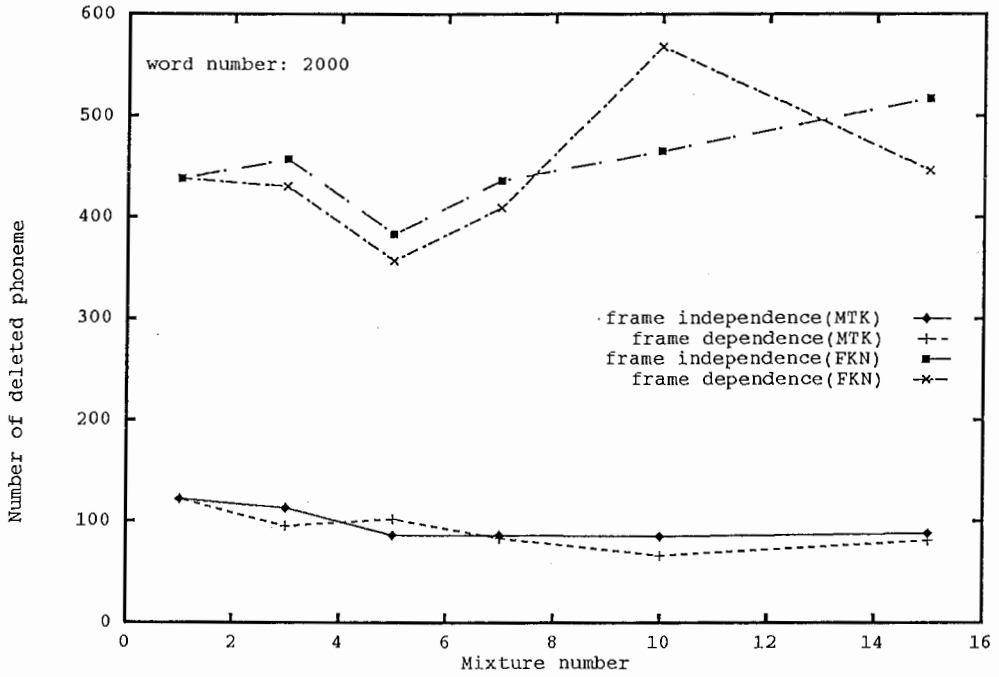

図 14: MTK による学習の単語認識での deletion 誤り結果

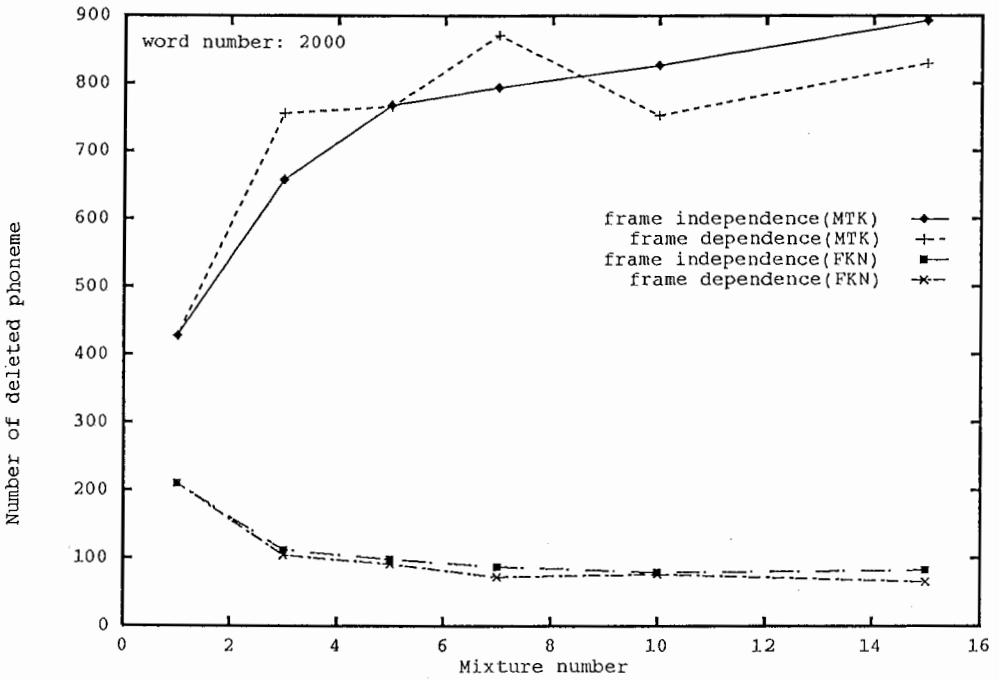

図 15: FKN による学習の単語認識での deletion 誤り結果

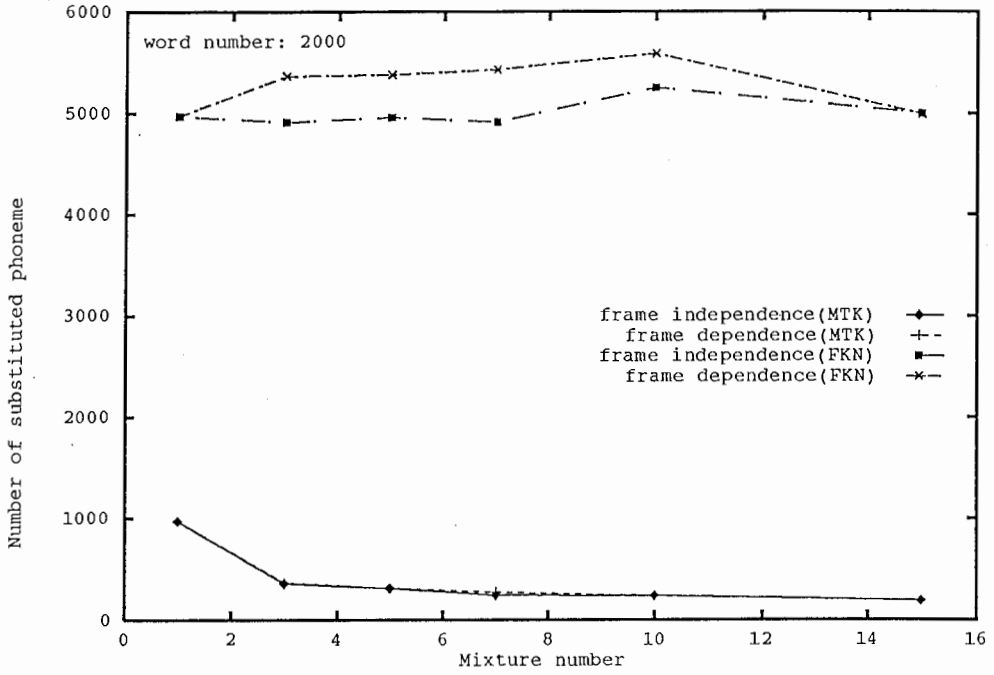

図 16: MTK による学習の単語認識での substitution 誤り結果

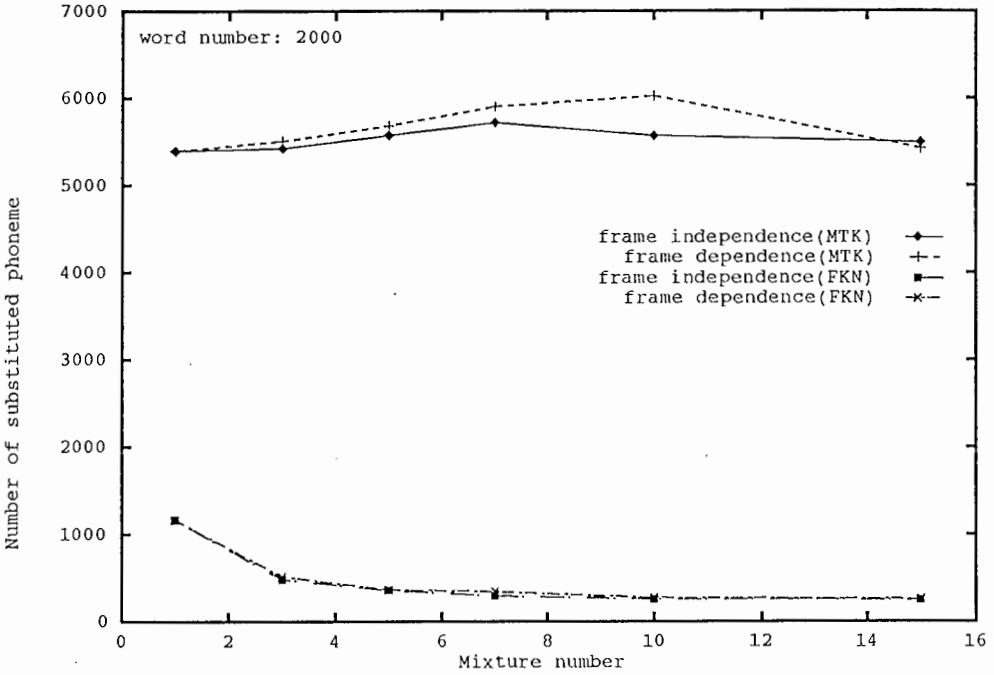

図 17: FKN による学習の単語認識での substitution 誤り結果

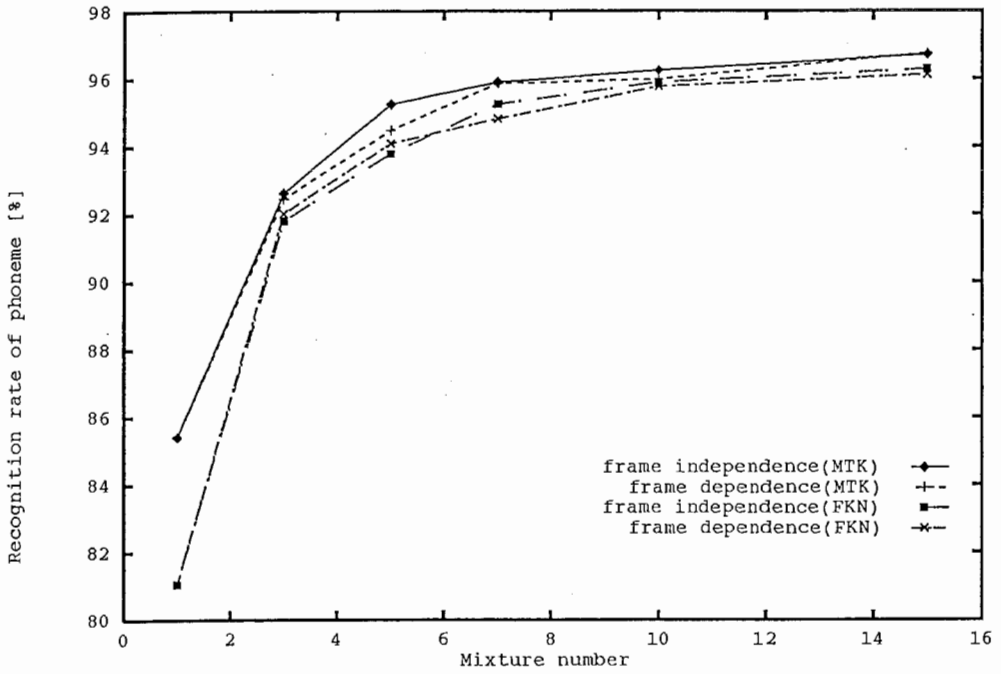

図 18: MTK と FKN による学習の音素認識結果

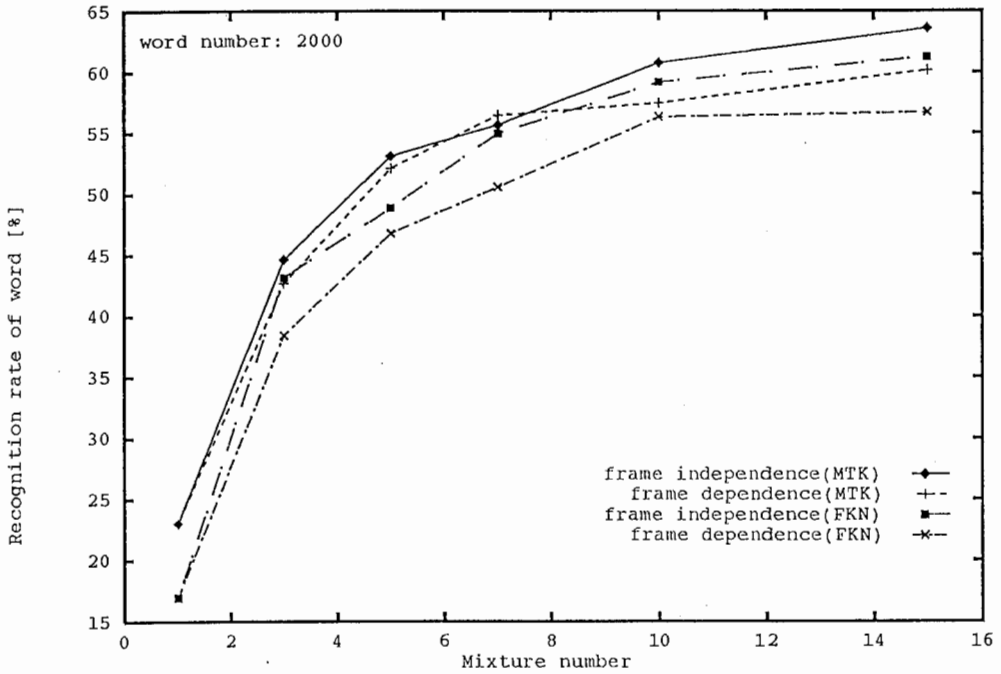

図 19: MTK と FKN による学習の単語認識結果

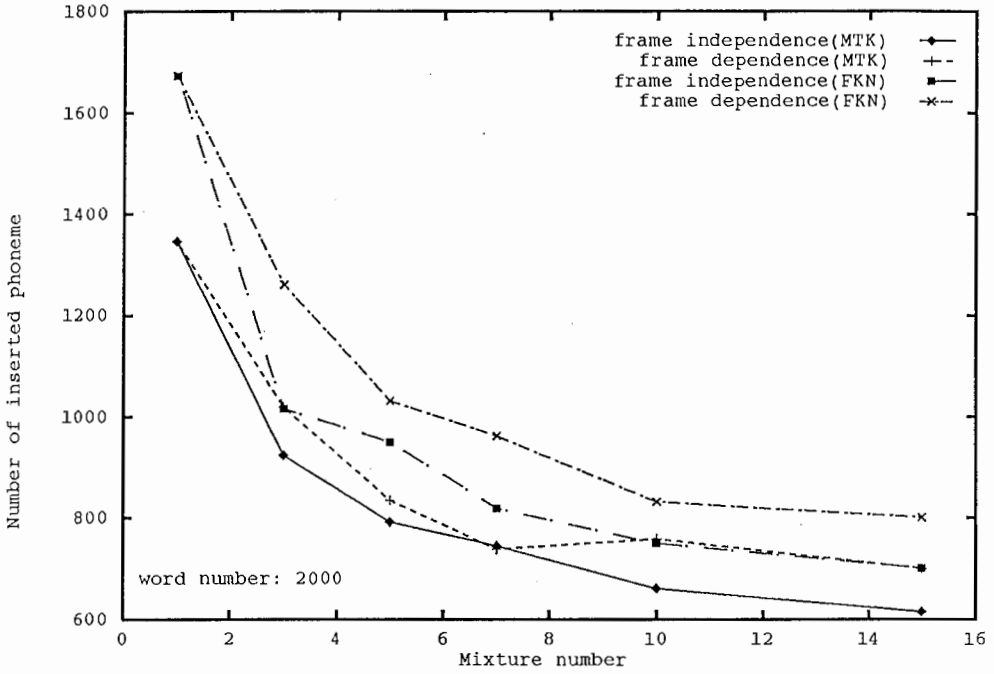

図 20: MTK と FKN による学習の単語認識での insertion 誤り結果

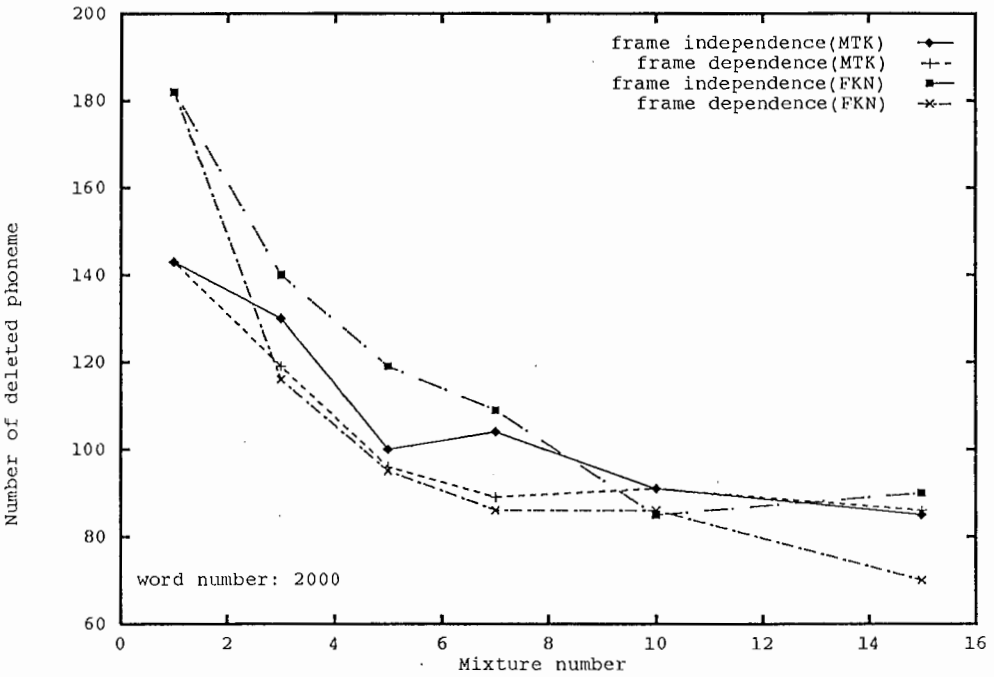

図 21: MTK と FKN による学習の単語認識での deletion 誤り結果

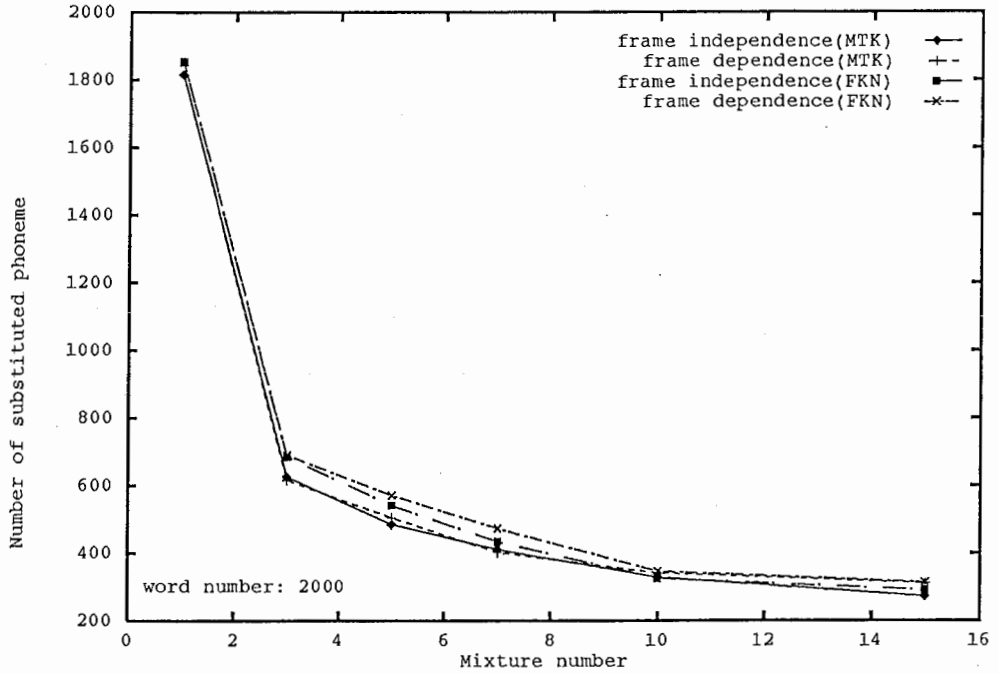

図 22: MTK と FKN による学習の単語認識での substitution 誤り結果

音声学習	音素認識				単語認識		
8 speakers	“MTK”	“FKN”	“MXM”	“FMS”	“MXM”	“FTK”	“FYM”

音素認識の結果を図 23に示す。

また、単語認識の結果を図 24に示す。単語認識での insertion 誤り、deletion 誤り、substitution 誤りの関係を図 25、図 26、図 27に示す。図 28に 8 人の誤りを合計した結果を示す。

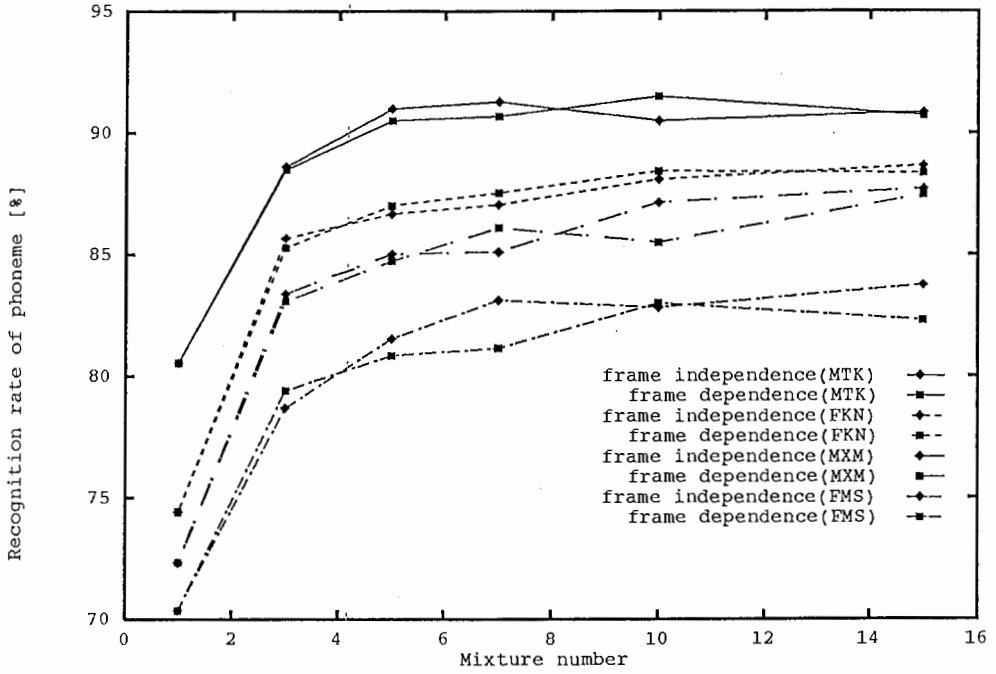

図 23: 8人の話者による学習の音素認識結果

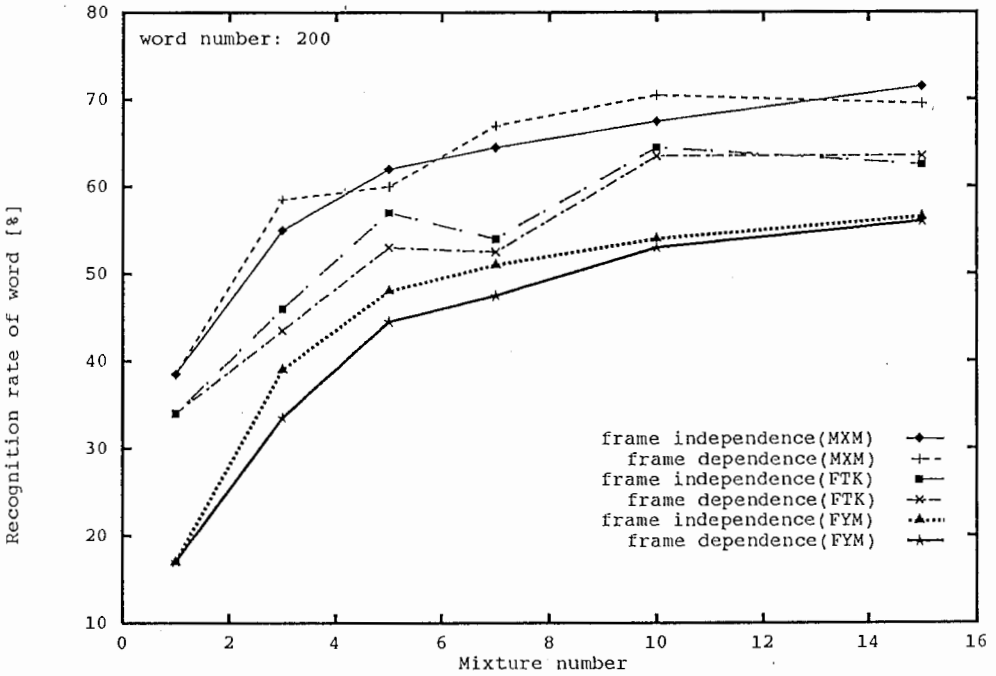

図 24: 8人の話者による学習の単語認識結果

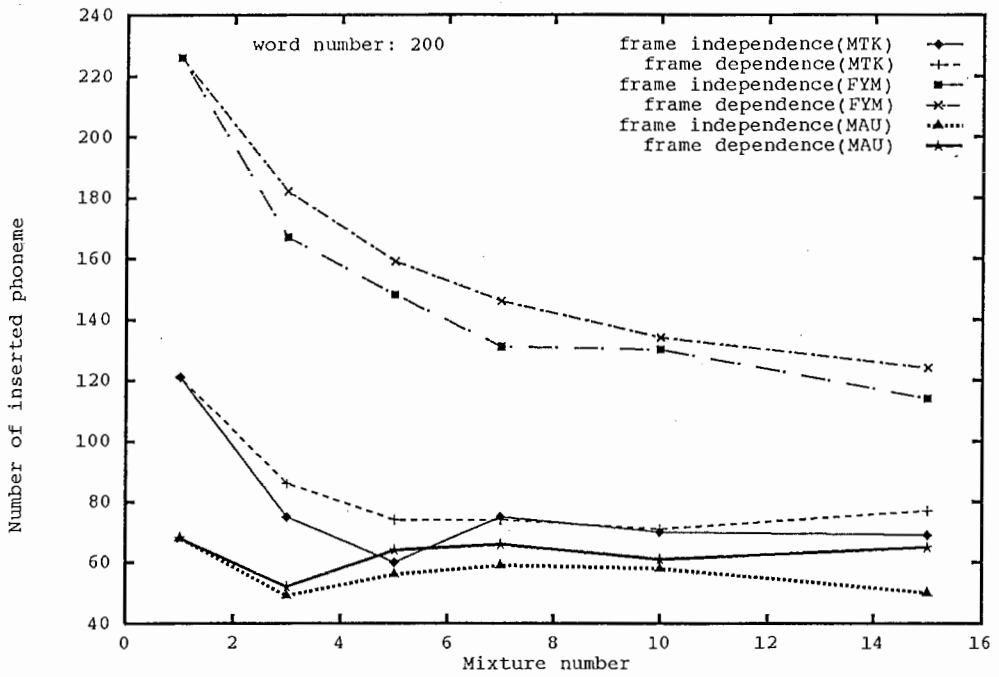

図 25: 8 人の話者による学習の単語認識での insertion 誤り結果

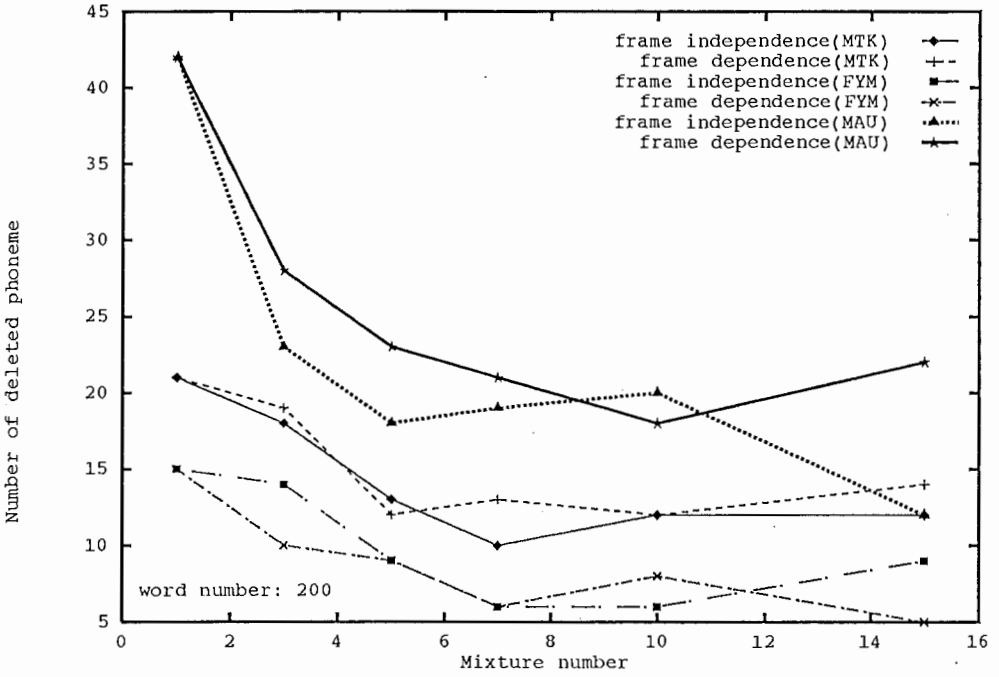

図 26: 8 人の話者による学習の単語認識での deletion 誤り結果

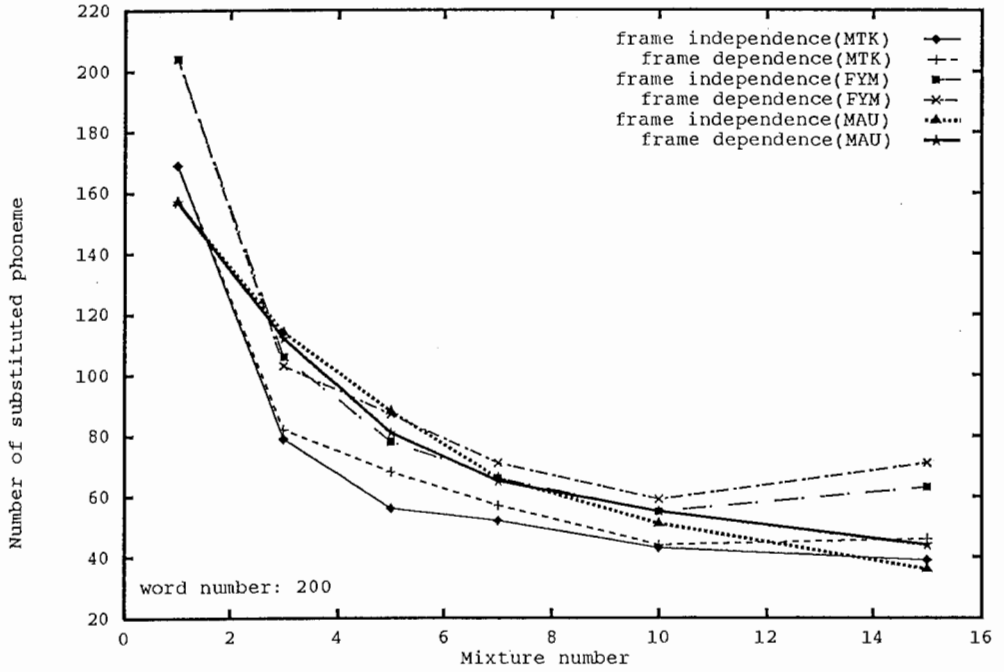

図 27: 8 人の話者による学習の単語認識での substitution 誤り結果

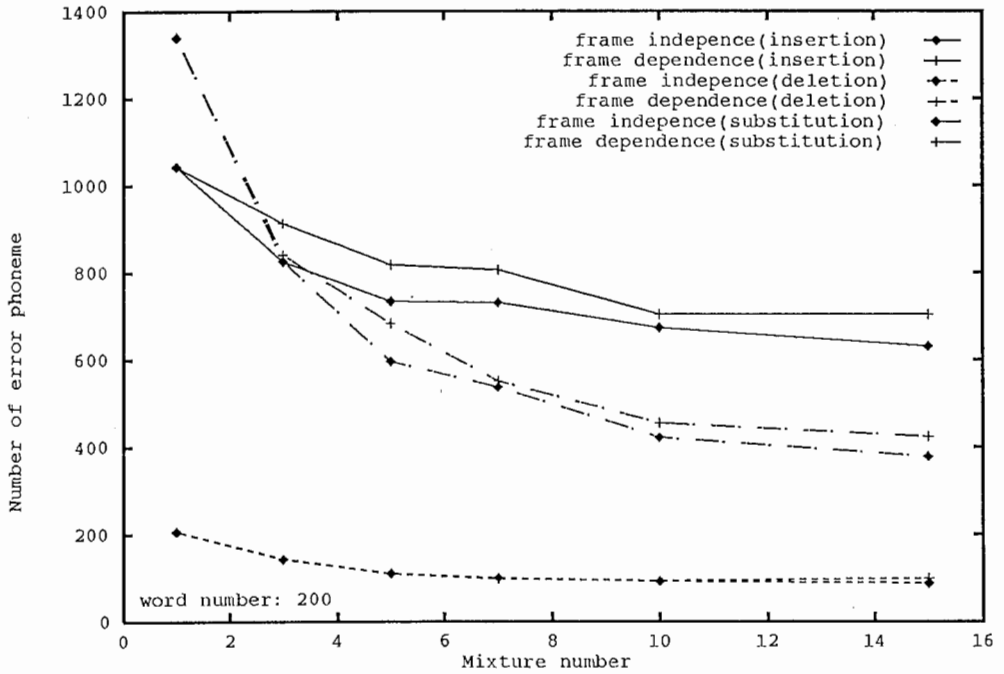

図 28: 8 人の話者による学習の単語認識での ins.,del.,sub. 誤り結果 (8 人の合計)

6　まとめ

音素認識では、混合数によっては、本方式が従来法よりも改善するところがあった。しかし、改善率はわずかである。傾向としては、学習話者が多い方が改善される割合は高い。

単語認識では、学習話者が１人のときは全く改善されず、学習話者が多くなるにつれて改善される割合は高くなった。単語認識において、insertion 誤り、deletion 誤りを見ると insertion 誤りは従来法より増加する傾向となり、deletion 誤りは従来法より減少する傾向となった。これは、時間フレームの依存性を考慮したことによる特徴であるといえる。音声データは１フレームの違いでは極端な変化は見られないのが普通である。よって、過去のフレームを用いて学習することにより、HMnet が急激な変化はないということを学習するはずである。しかし、学習の音声データは音素に分けるために強制的に分割されている。その分割によって、音声は連続でなくなり、HMnet はそれを学習している。その影響として insertion 誤りが増加したと考えられる。

このように本方式での音声認識は、insertion 誤りが増加するなどの特徴は見られたが、認識率としては、従来法とあまり変化は見られなかった

今後の課題として、音声データにノイズが含まれていると、１フレーム前と現在のフレームに大きな違いが含まれることになる。したがて、過去のフレームをそのまま用いるのではなく、過去 2-3 フレームを平均した音声データを用いるなどの検討が考えられる。また、現在は音声学習で音声データにラベルの区切りの情報を与えている。しかし、音素の区切りとは一般に明確なものではない。徐々に変化していくものである。したがって、音声学習で音素の順番だけを与え区切りを与えないで学習できるようにする。このようにすれば、本方式ではその音素が変化する遷移をも学習することになるので認識率が上がるのではないかと考えられ、その検討が考えられる。

参考文献

[1] 鷹見 淳一, 嵯峨山 茂樹: “逐次状態分割法による隠れマルコフ網の自動生成”, 電子情報通信学会論文誌 (D-II), Vol.J76-D-II, No 10, pp.2155-2164,(1993).

[2] G. D. Forney: *“The Viterbi algorithm”*, Proc. IEEE,61,pp268-278(1973).

[3] L. R. Rabiner, B. H. Juang: *“Fundamentals of Speech Recognition”*, Prentice Hall,pp329-389(1993).

[4] C. J. Wellekens: *“Explicit Time Correlation in Hidden Markov Models for Speech Recognition”*, ICASSP, pp.384–386(1987)

[5] Satoshi Takahashi, Tatsuo Matsuoka, Yasuhiro Minami, Kiyohiro Shikano: *“Phoneme HMMs constraned by Frame Correlations”*, ICASSP, pp.II 219-222(1993)

[6] 鷹見 淳一: *“SSS-ToolKit(Ver.3.0) User's Manual”*, ATR Interpreting Telecommunications Research Labs.,(1993).

付録 A　expand_sample の追加オプション

expand_sample [-fd *NUMBER*] [-fm *NUMBER*]

-fd: 使用する過去のフレーム数を指定する。従来法では 0、本研究の方式では 1、となる。指定されなかった場合は 0 が設定される。

-fd: 過去のフレームに対して何フレーム平均するかを指定する。従来法および本研究では、1 を指定する。指定されなかった場合は 1 が設定される。

設定例:

expand_sample -di 34 -fd 0

音声データ 1 フレーム 34 次元、現在のフレームのみを出力。出力データは 34 次元。出力される値:

$$O_t$$

expand_sample -di 34 -fd 1

音声データ 1 フレーム 34 次元、現在のフレームと過去の 1 フレームを出力。出力データは 68 次元。出力される値と順番:

$$O_{t-1}, O_t$$

expand_sample -di 34 -fd 1 -fm 2

音声データ 1 フレーム 34 次元、現在のフレームと過去の 1 フレームを出力。ただし、過去のフレームはそのフレームとそれより 1 フレーム前を平均したものとなる。出力データは 68 次元。出力される値と順番:

$$\frac{O_{t-2} + O_{t-1}}{2}, O_t$$

expand_sample -di 34 -fd 2 -fm 2

音声データ 1 フレーム 34 次元、現在のフレームと過去の 2 フレームを出力。ただし、過去のフレームはそのフレームとそれより 1 フレーム前を平均したものとなる。出力データは 102 次元。出力される値と順番:

$$\frac{O_{t-3} + O_{t-2}}{2}, \frac{O_{t-2} + O_{t-1}}{2}, O_t$$

expand_sample -di 34 -fd 2 -fm 3

音声データ 1 フレーム 34 次元、現在のフレームと過去の 2 フレームを出力。ただし、過去のフレームはそのフレームとそれより 2 フレーム前, 1 フレーム前を平均したものとなる。出力データは 102 次元。出力される値と順番:

$$\frac{O_{t-4} + O_{t-3} + O_{t-2}}{3}, \frac{O_{t-3} + O_{t-2} + O_{t-1}}{3}, O_t$$

付録 B Exe.retrain_HMnet,Exe.recognize_HMnet, Exe.typewriter_HMnet の追加オプション

Exe.retrain_HMnet [-fd *NUMBER*]
Exe.recognize_HMnet [-fd *NUMBER*]
Exe.typewriter_HMnet [-fd *NUMBER*]

-fd: 使用する過去のフレーム数を指定する。従来法では 0、本研究の方式では 1、となる。指定されなかった場合は 0 が設定される。

設定例:

Exe.retrain_HMnet -di 34 -fd 0 -if in_file -of out_file

音声データ 1 フレーム 34 次元、現在のフレームのみで学習

Exe.retrain_HMnet -di 68 -fd 1 -if in_file -of out_file

音声データ 1 フレーム 34 次元、現在のフレームと過去の 1 フレームで学習

Exe.recognize_HMnet -di 34 -fd 0 -if in_file -of out_file

音声データ 1 フレーム 34 次元、現在のフレームのみで認識

Exe.recognize_HMnet -di 68 -fd 1 -if in_file -of out_file

音声データ 1 フレーム 34 次元、現在のフレームと過去の 1 フレームで認識

Exe.typewriter_HMnet -di 34 -fd 0 -if in_file -of out_file

音声データ 1 フレーム 34 次元、現在のフレームのみで typewriter 認識

Exe.typewriter_HMnet -di 68 -fd 1 -if in_file -of out_file

音声データ 1 フレーム 34 次元、現在のフレームと過去の 1 フレームで typewriter 認識