한국어 숫자음 인식을 위한 TDNN과 HMM의 결합방법에 관한 연구

서원택°, 조범준
조선대학교 컴퓨터공학과

The Study on the Integration method using TDNN and HMM for Korean Digit Speech Recognition

Wontaek Seo, Beomjoon Cho
Dept. of Computer Engineering, Chosun Univ.
E-mail : wontagi@ai.chosun.ac.kr, bjcho@mail.chosun.ac.kr

요 약

본 논문에서는 한국어 숫자음 인식을 위한 시간 지연 신경망(Time delay neural network-TDNN)과 은닉 마르코프 모델(Hidden Markov Model-HMM)의 결합 방법에 대해서 연구하였고 그 성능을 측정하였으며, 기존의 시스템과 비교 평가하였다. 이 알고리즘은 TDNN과 HMM의 구조적인 결합에 기반하고 있는데 TDNN의 두번째 은닉층의 출력이 HMM의 입력으로 들어가도록 구성되었다. 그러면 HMM은 TDNN의 출력으로 각 단어에 대해서 훈련과정을 거치게 된다. 이렇게 구성된 인식알고리즘은 TDNN의 뛰어난 단기간(Short-time)분류 기능과 HMM의 시간 정렬(time-warping) 능력을 동시에 갖게 된다. 위의 과정을 컴퓨터 시뮬레이션을 이용하여 구현하였으며, 한사람의 음성을 녹음하여 실험한 결과 기존의 TDNN만으로 만들어진 인식기보다는 3%, HMM만으로 구성된 인식기 보다는 5.7%나은 성능을 얻을 수 있었다.

1. 서론

인간의 가장 자연스러운 의사전달의 수단으로 음성을 꼽을 수 있다. 이런 이유로 음성을 이용하여 각종 기계나 도구들을 쉽게 조작하고자 하는 것은 인류의 오랜 꿈 중의 하나였다. 공상과학 영화를 보면 인간과 컴퓨터가 자연스럽게 이야기 하는 장면을 자주 볼 수 있는것도 이런 환경을 항상 인간이 꿈꾸고 있기 때문일 것이다. 그러나 사람들의 기대와 상상과는 달리 아직 우리가 영화에서 보는 바와 같이 자유스러운 대화를 하는 컴퓨터는 등장하지 못했다. 만약에 이러한 기술이 개발된다면 기술적인 파급효과는 물론 경제적, 사회적인 면에서도 획기적일 것이다.

음성인식의 궁극적인 목표는 잡음이 있는 실제 환경하에서 임의의 화자가 어휘에 제한없이 자연스럽게 발음한 연속 음성을 실시간에 인식 및 이해하는 것이라고 할 수 있다. 위와 같은 목표를 이루려고 할 때 고려해야 할 사항은 크게 다음의 다섯가지로

나타내 볼 수 있다.

i. 화자 독립성(Speaker independence)
ii. 발음 방법 또는 속도
iii. 인식대상어휘의 난이도
iv. 언어의 문법구조 및 주제
v. 음성통신 환경

그러나, 특정화자의 음성만을 대상으로 하여 수십 단어 정도의 제한된 어휘를 또박또박 띄어 발음한 음성을 인식하는 것과 같이 제한된 목표를 설정할 경우, 매우 우수한 성능을 나타내는 음성인식 시스템들이 개발되고 있다[1][2].

음성인식 시스템을 구현하는데 있어서 고려되어야 할 사항으로는 음성 특징 추출 방법, 인식단위 설정 및 단어 인식 방법 등 각 부분에 대한 알고리즘을 설정하는 것이다.

본 논문에서는 신경망의 패턴 분류능력과 HMM의 시간처리능력의 장점을 결합할 수 있는 방법에 대한 연구와 실험결과를 제시한다. 2장에서는 입력된 음성에서 인식기에 실제로 입력될 특징을 추출하는 과정을 설명하고, 3장에서는 특징 추출과정에서 추출된 특징벡터를 이용하여 실제로 음성을 인식해 내는 인식 알고리즘에 대해서 설명한다. 4장에서는 실험 및 고찰을 하였으며 5장에서는 결론을 맺는다.

2. 음성특징추출 과정

인간의 음성은 아날로그 신호로서 나타나는데 이러한 아날로그 신호를 microphone 을 이용하여 시스템으로 받아들이면 신호는 전기적인 연속파형으로 된다[3]. 이렇게 받아들여진 전기적인 파형을 그대로 인식 시스템에 입력하기에는 양이 너무 방대하기 때문에 원신호의 특징을 최대한 손실하는 않고 음성인식 시스템에 적합한 유용한 형태의 신호로 표현하기 위한 과정을 거치는데 그것을 음성 특징 추출과정이라고 한다.

연속 파형은 먼저 전처리 과정(preprocessing)을 통과하는데 Low pass filtering, Sampling, A/D 변환등과 음성신호를 묵음 구간으로부터 분리해 내는 끝점 검출 과정이 여기에 포함된다[4]. 전처리 과정을 통과한 음성 신호는 특징 추출과정에 들어가 특징 파라미터를 구하게 된다. 음성특징 추출의 대표적인 방법으로는 filter bank의 출력을 이용하는 방법, LPC(Linear Predictive Coefficients)를 이용하는 방법, cepstrum 계수를 이용하는 방법등이 있다. 이들중 LPC는 모음에 대해서는 비교적 정확히 모델링되지만, 자음과 비음에 대해서는 성능이 저하되는 단점이 있는 것으로 알려져 있다. 그래서 이 방법을 그대로 사용하지 않고 LPC계수들로부터 cepstrum 계수를 구하고 이에 적절한 가중치를 가하면 잡음에서도 강인한 음성 특징을 구할 수 있다. 본 논문에서는 음성특징 파라미터로 12차 LPC based mel-scaled frequency cepstrum 계수를 사용하였다. 파라미터열로 변환된 음성신호는 training, simulation 그리고 실제 인식과정에서 모두 사용된다.

2.1 전처리

전처리 과정은 특징 벡터를 추출하기 위한 첫 단계로서, 먼저 음성 신호로부터 고주파 성분의 잡음을 제거하기 위하기 위하여 4.5 kHz 의 cut-off 주파수를 갖는 low pass filtering을 하였다. 이러한 신호를 10kHz sampling rate 을 갖는 16 bit A/D Converter로 양자화한 후, 실 음성 구간을 얻어내기 위해서 끝점 검출 과정을 거치게 한다. 음성구간 검출의 정확성 여부는 음성인식의 정확도에 큰 영향을 미치기 때문에 정확한 검출이 필요하다. 또한 전체 계산량에도 영향을 미치기 때문에 실시간 시스템에 사용하기 위해서는 계산량을 증가시키지 않는 효율적인 방법이어야 한다. 간단하게 끝점검출과정을 보자면, 먼저 묵음이라고 가정되어진 몇 개의 구간으로부터 잡음의 통계적 특성 구하고 이를 이용하여 threshold를 구한다. 그런 다음 단구간 에너지(Short-term Energy)와 비교하여 대략적인 음성구간을 결정하고, 영교차율(Zero-Crossing rate)을 이용하여 정확한 음성구간을 구한다. 끝점 검출 과정을 통해 얻어진 음성 데이터는 pre-emphasis 과정을 거치게 되는데 이는 음성 신호에 포함되어 있는 직류성분이나 저주파의 잡음을 제거 시키며 dynamic range를 감소시키고, 성대의 spectral shape를 더욱 정확히 표현할 수 있도록 하는 효과를 갖는다. 아래

의 그림은 전처리 과정을 나타내었다.

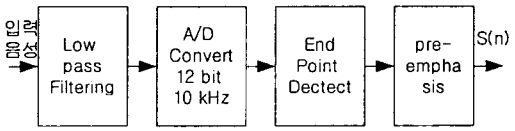

그림 2-1. 전처리 과정 블록도

2.2 LPC coefficents

선형 예측 분석을 사용하면 음성 신호, 혹은 음성 스펙트럼이 가진 특성을 상대적으로 적은 수의 파라미터만으로 정확하게 표현할 수 있다는 장점이 있고, 또한 선형 분석 자체가 많은 연산량을 요구하지 않는다는 장점도 있다.

시간 t에서의 음성 샘플이 x_t라고 한다면 선형 예측분석법에서는 현재의 음성 샘플을 이전 p개의 샘플로부터 예측을 한다. 이때 예측값과 실제값과의 차이를 e_t라고 하면 다음과 같은 식이 성립한다.

$$x_t = -\sum_{i=1}^{p} \alpha_i x_{t-i} + e_t \qquad (2.1)$$

위식에서 α_t가 구하고자 하는 선형 예측 계수이고, e_t를 잔차신호라고 한다. 선형 예측 분석은 시간 t에서의 음성 샘플이 이전 p개의 음성 샘플로부터 효과적으로 예측될 수 있다는 가정에 기반 한다. $X(z)$와 $E(z)$를 각각 원신호와 오차신호의 z-변환이라 한다면 다음식과 같이 표현할 수 있다.

$$X(z) = \frac{1}{A(z)} E(z), \quad A(z) = 1 + \sum_{i=1}^{p} \alpha_i z^{-i} \qquad (2.2)$$

위의 식 2.2 를 보면 오차 신호를 $1/A(z)$에 통과시키면 원음을 얻을 수 있으므로 $1/A(z)$를 선형 예측 필터(linear prediction filter)라고 하며 원음을 $A(z)$에 통과시키면 오차신호를 얻을수 있으므로 $A(z)$를 역필터(inverse filter)라고 한다. $A(z)$와 같은 형태를 all-pole 모델이라고 하며 음성은 선형 예측 분석법의 차수인 p가 10이상인 경우 all-pole 모델에 의해서 효과적으로 표현될수 있음이 알려져 있다. p 차 all-pole 필터의 전달함수는 다음과 같이도 나타낼수 있다.

$$H(z) = \frac{1}{1 + \sum_{i=1}^{p} \alpha_i z^{-i}} \qquad (2.3)$$

2.3 켑스트럼

컵스트럼은 로그 크기 스펙트럼(logarithmic amplitude spectrum)의 역 퓨리에 변환(inverse Fourier transform)으로 정의된다. 켑스트럼이 가진 가장 큰 특징이라고 하면 음성이 갖는 정보에서 스펙트럼 포락의 정보와 세부 구조 정보를 분리해 낸다는 것이다.

LPC 켑스트럼 계수를 얻기 위한 식은 다음과 같다

$$\hat{c}_1 = -\alpha_1$$

$$\hat{c}_n = -\alpha_n - \sum_{m=1}^{n-1} \left(1 - \frac{m}{n}\right) \alpha_m \hat{c}_{n-m} \quad (1 < n \le p) \qquad (2.4)$$

$$\hat{c}_n = -\sum_{m=1}^{p} \left(1 - \frac{m}{n}\right) \alpha_m \hat{c}_{n-m} \qquad (p < n)$$

식 2.4 에 의해서 얻어지는 켑스트럼 계수는 선형 예측 계수로부터 얻어지기 때문에 LPC 켑스트럼 계수라고 불린다.

본 논문에서는 30msec 크기의 프레임에 hamming window를 사용하였고 12차의 autocorrelation method로부터 LPC계수를 구하여 식 2.4로 conversion한 LPC cepstral coefficient를 구하였고 bilinear transform을 거쳐서 12차 LPC-based mel-cepstral cofficient 를 입력벡터로 사용하였다.

3. 음성인식 알고리즘

이번 장에서는 개발된 음성인식 알고리즘중 가장 널리 사용되고 있는 TDNN과 HMM에 대해서 살펴보고 TDNN과 HMM을 결합하는 알고리즘에 대해 설명하겠다..

3.1 TDNN

신경회로망을 이용하여 음성을 인식하는 방법이 기존의 패턴 인식 문제에 적용되었던 방법들이나 음성 인식 모델에 비해서 가지는 큰 장점으로는 학습으로 인한 일반화(Generalization)과 새로운 자료의 추가시 학습과정이 다른 방법에 비해서 간단하다는

것과 병렬계산 능력(Parallel processing), 그리고 오류 감수(fault tolerance)등을 들 수 있다. 그러나 신경 회로망 자체에 시간적인 개념을 부여하기가 어려우며, 기존의 분석 방법을 통해서 얻어진 음성에 대한 지식을 신경 회로망에 표현할 수 있는 방법의 부재등으로 인하여 신경 회로망을 이용하여 음성을 인식하는 데는 많은 어려움이 있다. 이러한 문제를 해결하기 위해서 TDNN(Time Delay Neural Network)이란 새로운 방법이 Waibel 에 의해 제안되었으며, 음성과 같은 동적인 패턴을 다루기 위해 지연요소, 시간적으로 처리하는 요소등을 이용해 입력 패턴에 내재되어 있는 시간적인 특징을 인식하는 신경망이다.

TDNN은 연속되는 음성구간 사이의 spectral구조를 학습할 수 있는 다층 신경 회로망으로 되어 있다. 각 층에서의 출력은 일정한 시간 간격을 두고 이루어지며, 입력층에서 출력층으로 감에 따라 출력 시간 간격을 늘려 처리하고자 하는 단위에 맞춤으로써 동적인 패턴의 특성을 위치와 무관하게 감지할 수 있는 특성이 있다.

다음 그림에서 TDNN의 기본 구조를 나타내었다

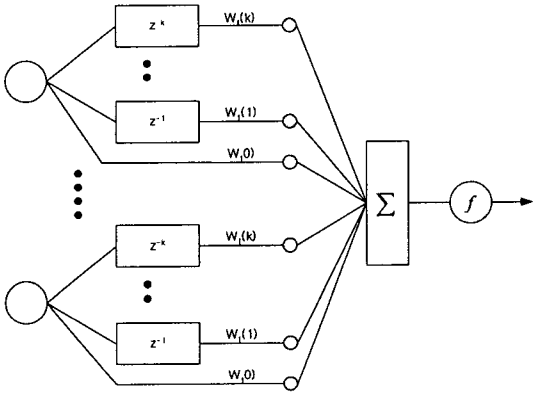

그림 3-1. TDNN의 기본구조

3.1.1 TDNN의 학습 알고리즘

TDNN의 학습 과정은 가장 널리 이용되고 있는 오류역전파(Back Propagation) 학습 알고리즘을 이용한다. 이 BP 알고리즘은 steepest descent rule을 사용하여 mean-square error(MSE)가 최소가 되도록 연결강도를 조정하는 학습법이다.

연결강도를 변경하는 과정을 간단하게 살펴보면 먼저 다음식에 의해 오류를 구하고 난뒤 각층에

$$E = \frac{1}{2}(d-y)^2 + E \qquad (3.1)$$

전파될 값을 다음식에 의해 구한다. 그리고 나서

$$\delta_y = (d-y)(1-y^2) \qquad (3.2)$$

$$\delta_z = \frac{1}{2}(1-z^2)\sum_{i=1}^{m}\delta_y w \qquad (3.3)$$

연결강도를 다음식에 의해 변화시킨다.

$$w^{k+1} = w^k + \Delta w^k = w^k + \alpha\delta_y z^k \qquad (3.6)$$

$$v^{k+1} = v^k + \Delta v^k = v^k + \alpha\delta_z x^k \qquad (3.7)$$

위의 과정들이 error가 허용할 수 있는 최소값에 도달할 때까지 계속 반복하게 된다.

TDNN 은 이러한 back-propagation 알고리즘을 이용하여 다음의 학습 과정을 통하여 학습된다.

1) 각 time-shifted window에 대한 weight들은 서로 같도록 제한한다.
2) 우선 각각의 time-shifted window들에 대해서 back-propagation 알고리즘을 적용하여 weights 의 변화량을 계산한다.
3) 그 변화량들의 평균값으로 각 weight들을 조정한다.

본 논문에서는 back-propagation의 학습속도를 향상시키기 위해 모멘텀 BP 알고리즘과 적응적 학습률을 동시에 적용하였다.

3.3 HMM

HMM은 실제적인 관측을 통해서 변화되는 통계적인 특징을 확률적으로 모델링하기 위하여 마르코프 과정을 이용한다. 각 상태열은 은닉되어 있고, 다른 관측 가능한 확률적인 과정들의 집합을 통해서만 관측될 수 있다. 이 모델의 각 은닉 상태 또는 각 상태간의 천이는 이산 또는 연속 확률 밀도 함수로 표현되는 출력 확률 분포 집합과 연관되어 진다. 그리고, 은닉 상태열과 관측 가능한 확률 과정 사이의 장막은 출력 확률에 의하여 표현된다.

일반적으로 HMM은 $\lambda=(A,B,\pi)$ 로 간결한 표시로 표현되며, 상태 수 N 과 경우수 M , 그리고 상태 천이 확률 A , 상태에 대응되는 출력확률 B , 초기 확률π의 세가지 확률 분포가 필요하다. 이 확률 변수들은 결국 $P(O|\lambda)$를 구하기 위한 것이다.

HMM을 실제 유용하게 사용하기 위해서는 다음과 같은 세 가지 중요한 문제를 해결해야 한다.

첫번째는 문제는 평가 문제(Evaluation Problem)로 관측열 $O=O_1,O_2,\cdots,O_T$과, 모델 $\lambda=(A,B,\pi)$가 주어지면, 모델에 의해 생성되는 관측열에 대한 확률 $P(O|\lambda)$을 어떻게 계산할 것인가의 문제이다. 이 문제를 해결하는 방법으로 전후향 알고리즘이 주로 사용되는 이 알고리즘에서 사용되는 전향 변수는 다음과 같이 정의된다.

$$\alpha_t(i)=P(O_1,O_2,\cdots,O_t,s_t=i|\lambda) \quad (3.8)$$

위의 식은 모델 λ가 주어지면, 시간 t에 도달하는 상태 i에서 t 시간까지의 부분 관측에 대한 확률을 나타낸다.

두번째 문제는 예측 문제(Estimation Problem)으로 주어진 관측열 O에서 $P(O|\lambda)$를 최대화하는 모델 변수 $\lambda=(A,B,\pi)$를 조정하는 문제이다. 따라서 이 문제는 관측들이 어떻게 출현되는 가를 가장 잘 표현하도록 모델 변수들을 최적화 시키는 문제이다. 이 문제를 해결하는 방법으로 Viterbi 알고리즘을 많이 사용한다.

세번째 문제는 해석 문제(Decoding Problem)로 관측열 O가 주어지면, 최적화 기준에 따라 가장 가까운 상태열 $S=s_1,s_2,\cdots,s_T$를 구하는 문제이다. 이 문제는 모델의 은닉된 부분을 복원시키는 과정이다. HMM에서 가장 어려운 문제는 주어진 모델에서 관측열의 확률을 최대화하도록 모델 변수 (A,B,π)를 조정하는 것이다. 이 문제를 maximum likelihood 모델 방법을 이용하여 분석적으로 풀 수 있는 방법은 없으며, 최적화 시키기 위해서는 반복적인 방법(iterative algorithm)이나 기울기 방법(gradient algorithm)을 사용하여야 한다. 반복적으로 수정하고 향상시키는 HMM 재예측(Reestimation)과정을 설명하기 위해서, 먼저 $\xi_t(i,j)$를 다음과 같이 정의한다.

$$\xi_t(i,j)=P(q_t=i,q_{t+1}=j|O,\lambda) \quad (3.32)$$

위식의 $\xi_t(i,j)$는 모델 λ와 관측열 O가 주어졌을 때 시간 t에 i상태에 있고 시간 $t+1$에는 j상태에 있을 확률을 나타낸다. 위의 식의 조건을 만족하는 그림을 나타내었다.

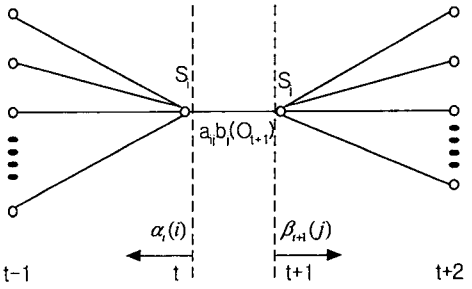

그림 3-2. 시간 t에 i상태이고 $t+1$시간에 j상태에 있을 확률의 결합관계(forward parameter 와 backward parameter와의 결합)

음성인식에서는 주로 left-to-right라는 모델을 사용한다. 이 left-to-right모델은 state 자체가 시간적 순서를 내포하고 있으므로 시간에 따른 음성의 특성을 효과적으로 모델링 하기에 적합하다.

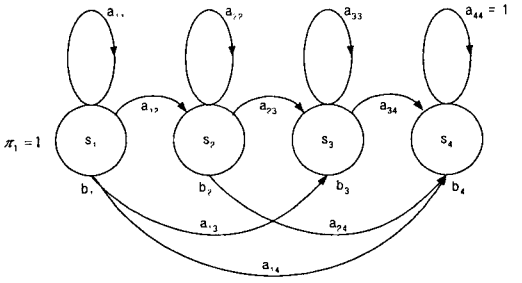

그림 3-3. 4 states left-to-right HMM

3.4 TDNN과 HMM을 결합하는 알고리즘

본 논문에서는 TDNN의 장점과 HMM 의 장점을 결합할 수 있는 알고리즘에 대해서 연구해 보았다. TDNN의 second hidden layer에서 출력되는 값들을 HMM의 관측열(object sequence)로 입력하여 분류해내는 알고리즘이다. 이러한 알고리즘은 인식 단어에 대해 그 unit들의 출력값을 단순히 합하여 인식하지 않고 이를 어떤 단어에 속할 membership value로 이용하

으로 TDNN만을 이용하는 알고리즘보다 효과적이다. 또한 단어 이하의 인식 단위로 학습된 모델을 가지고 단어 모델로 쉽게 확장시킬 수 있기 때문에 많은 수의 격리단어의 인식에도 쉽게 이용될 수 있다. TDNN과 HMM 의 결합방법에 대한 그림을 나타내었다.

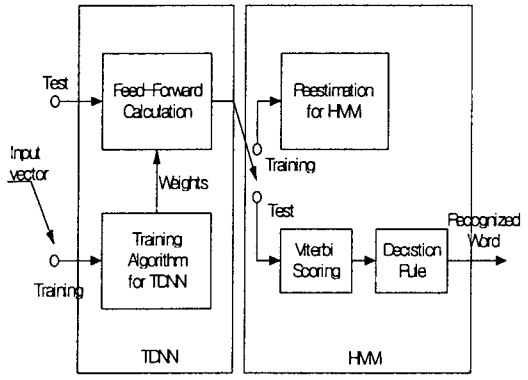

그림 3-4. TDNN/HMM 알고리즘의 구성도

4. 실험 및 고찰

본 장에서는 TDNN/HMM을 이용한 한글 숫자음 인식 알고리즘의 성능을 평가하기 위한 computer simulation 및 이의 결과에 대해서 고찰해 보도록 하겠다

실험을 위해서 3개의 단어군 음성을 생성하였다.

단 어 군	단 어
단어 1군	시작, 끝
단어 2군	영,일,이,삼,사,오,육,칠,팔,구
단어 3군	공,하나,둘,삼,넷,오,여섯,칠,팔,아홉

표 1. 실험에 사용된 단어군

위 단어군을 한명의 화자가 20번씩 녹음하여 훈련용 데이터와 실제인식 실험에 사용하였다.

단어 1군을 위한 인식실험에서는 TDNN 을 기반으로하는 인식시스템을 구성하고 실험하였다. 5개의 음성은 훈련용 데이터로 사용하였고 15개는 인식용 데이터로 사용하였다. 그 결과 100%의 인식결과를 얻었는데 이는 두 단어가 발음상 유사성이 거의 없기 때문이라고 판단된다.

단어 2군에 대한 실험으로는 TDNN의 시스템, HMM의 시스템, 그리고 TDNN과 HMM의 시스템을 구현하여 비교하여 보았다 그 결과 TDMM의 경우 오류율이 8%, HMM의 경우 10.7%, TDNN과 HMM을 결합한 알고리즘은 5%의 오류율을 얻었다. 결국 TDNN 보다는 3%, HMM보다는 5.7%의 인식률 향상을 얻을 수 있었다.

단어 3군에 대한 실험에 사용된 음성 데이터는 단어 2군에 비해서 발음상 유사성이 더욱 적은 단어들고 구성되어 있어 단어 2군에 사용했던 대로 적용을 해 본 결과 3%더욱 향상된 결과를 얻었다.

5. 결론

본 논문에서는 TDNN과 HMM을 결합하여 한국어 숫자음 인식을 할 수 있는 알고리즘을 연구하였다.

그결과 TDNN 의 경우 92%, HMM의 경우 89.3%, TDNN과 HMM을 결합한 경우 95%의 인식률을 얻음으로써 한가지만 사용한 경우보다 더 나은 성능을 얻을 수 있었다. 일반 숫자음의 경우에는 사람사이의 대화에서도 쉽게 구분할 수 없을 만큼 비슷한 발음이 존재하기 때문에 두번째 실험에서도 에러가 나타나는 것을 알 수 있었다. 그래서 세번째 실험에서 단어 3군을 구성할 때는 이런 발음상의 유사성을 제거한 숫자음을 사용하여 실험하여 보았는데 두번째 실험의 결과보다 훨씬 나은 결과를 얻을 수 있었다.

앞으로의 연구과제는 단어단위의 인식보다 더 작은 음소단위의 인식을 연구하여 연속음성 및 대용량의 인식으로 발전하는 것이다.

참고문헌

[1]S. E. Levinson and D. B. Roe, A perspective on speech recognition, IEEE Comm. Magazine vol. 28, pp. 28-34, Jan. 1990

[2] L. R. Rabiner and S. E. Levinson, Isolated and connected word recognition theory and selective application, IEEE Trans. Comm., vol. COM-29, pp. 621-659, May 1981.

[3] Oppenheim, A. V. and Schafer, R. W.. Digital Signal Processing, New Jersey, Pretice-Hall, 1975.

[4] 구명완외, "실시간 음성 끝점검출 알고리즘," 제 5 회 신호처리 합동 학술회 논문집, 제 5 권 1호, pp. 11-14, 1992