J. KSIAM **Vol.5**, No.2, 7-16, 2001

분별학습에 기반한 전화 숫자음 음성인식

한 문 성

요 약

음성인식 시스템에 있어서 현재 가장 널리 사용되고 있는 Hidden Markov Model(HMM)은 확률 모델을 기반한 것으로 데이터에 대한 통계처리를 학습과정으로 하고 있다. 한국어 연속 숫자음에 대한 음성인식은 고립 숫자음 인식과는 달리 충분한 학습데이터만으로는 만족할 만한 결과를 가져오지 못한다. 이 논문에서는 연속 숫자음 음성인식에 있어서 비슷하게 발음되는 숫자음과 같은 숫자에 대해 다양하게 발음되는 숫자음에 대해 HMM의 한계를 제시하고 그 해결책으로 Discriminant 학습의 적용방법을 제시한다. 연속 숫자음의 인식 시스템을 구현하는 데 있어서 인식률 낮은 부분에 Discriminant 학습을 적용하여 인식률을 대폭 향상시킨 실험결과를 제시한다.

제 1 절 서론

비밀번호, 구내 전화번호 자동연결과 같은 시스템의 구축에 있어서 필수적인 부분이 숫자음 인식이다. 또한, 고립 숫자음이 아닌 연속 숫자음의 인식시스템의 구축이 필요하다. 이러한 음성인식 시스템은 텔레뱅킹 시스템, 주식 정보 응답 시스템과 같은 금융서비스에 광범위하게 사용되고 있으나, 인식의 정확도에 있어서 아직 많은 개선이 요구되고 있다.

한국어 숫자음 음성의 경우, 연속적으로 발음하면 자음접변과 같은 음운현상이 나타나고, 특히 숫자음의 경우는 발음의 길이가 각기 달라서 음성인식 시스템 구축이 매우 어려운 편이다.

HMM은 Baker와 IBM의 연구자들에 의해 음성인식에 도입된 이래 음성인식 분야에서 가장 성공적인 방법으로 평가받고 있는 방법이다. HMM은 확률 모델을 기반으

로 한 학습방법으로 패턴의 시간축 정합등 음성에 적합한 여러 장점으로 인해 널리 사용되고 있으며, 현재까지 만들어진 음성인식 시스템 중에서도 가장 좋은 인식률을 나타내고 있다.

HMM은 음성신호를 Markov 상태열의 천이과정에서 발생하는 것으로 보는 확률모델을 기반으로 하여 음성 데이터로부터 추출된 정보를 통계학적으로 모델링하는 알고리즘이다. 근래의 음성인식 시스템들은 연속음성을 대상으로 인식을 수행하는 경우가 대부분이다. 따라서 음소단위의 학습과 인식 시스템의 구축이 요구되지만, 이것은 많은 작업과 인식하는 데 계산 량이 많은 단점을 가지고 있다.

네 자리 숫자 비밀번호, 구내 전화번호와 같이 실세계에서 쓰이지만 작은 범위에서 이용되는 분야에 적용되는 음성인식은 HMM을 이용한 간단한 시스템의 구축으로 충분히 사용될 수 있다. 그러나, HMM은 학습 시에 많은 학습데이터를 필요로 하고, 비슷한 음성 신호에 대해서는 인식오류율이 높은 단점을 가지고 있다. 고립단어 전화음성에 대해 HMM을 이용하기 위해 필요한 학습데이터는 양이 부족하면 인식률이 낮다. 실험을 통해 살펴보면 1-4음절의 단어 대해서는 약 70-100개 정도의 음성 데이터 수집으로 95%이상의 인식률을 가짐을 알 수 있다.

비슷한 발음의 음성 신호에 대한 처리, 잡음처리 등에 널리 쓰이는 방법은 Discriminant 학습이다 [1], [2]. 또한 Discriminant 학습은 적은 학습데이터가 필연적인 화자인식, 화자확인 분야에도 널리 적용되어 지고 있다 [5].

연속 숫자음의 발음을 잘라 HMM을 이용해 각각 고립 숫자음으로 인식하고, 낮은 인식률을 갖는 부분에 대해서는 Discriminant 학습을 적용한다면, 적은 비용과 계산 량으로 효과적인 음성인식 시스템을 구축할 수 있다.

제 2 절 한국어 연속 숫자음의 인식

연속 숫자음 인식에 있어서 많이 사용되는 방법은 고립단어 모델과 음소단위모델을 이용하는 것이다. 고립단어모델은 '일', '0|', '삼', ··· 과 같이 숫자음 각각을 인식단위로 삼는 것이며, 음소단위모델은 '칠'에 대해 ㅊ- ㅣ-ㄹ 과 같이 세 가지의 음소를 인식단위로 삼는 것이다.

음소단위 모델을 사용할 경우, 인식어휘에 비교적 독립적인 인식 시스템이 될 수 있으며 추가적인 인식모델을 설정할 때에 도 큰비용이 없이 가능하다. 그러나, 인식시스템의 구축에 많은 비용이 요구되며, 학습데이터에 대한 정확한 segmentation을 필요

로 하고 있다.

이 논문에서는 고립단어 모델을 기반으로 하고 있다. 따라서, 모든 숫자음을 인식단위로 설정할 뿐만 아니라 한국어 숫자음이 발음에 있어서 여러 가지 형태로 변이가 되기 때문에 이러한 발음을 모두 인식단위로 설정하야 한다.

숫자음의 경우 같은 숫자에 대한 발음이 숫자의 위치에 따라, 화자의 발음 습관에 따라 다 양하게 발음된다. 예를 들어,

- '일'의 경우, 일반적으로는 '일'로 발음되나, '육' 다음에 발음될 경우, '길'로 발음된다. 또한 이 경우 '육'의 경우도 '유'로 발음되는 경향이 많다. '일'은 '삼' 뒤에 발음될 경우 '밀'로 발음되며, '칠'이나 '일' 뒤에 발음될 경우에는 '릴'로 발음된다.

  고립단어 인식의 기본을 따르자면 네 가지 각각에 대한 학습을 진행하고, 인식결과를 모두 '일'로 처리해야 하지만, 많은 학습 데이터가 필요하고, 계산 량이 늘어난다는 점에서 바람직하지 않다. 따라서 '밀', '일', '길', '릴'의 네 가지 학습자료를 한꺼번에 이용한 HMM학습을 고려해야 한다.

- '이'의 경우, '0|', '리', '기', '미' 등 다양하게 발음되며, 특히, '0|0|' 의 경우와 같이 같은 숫자음이 발음될 경우, '0|' 인지 '0|0|' 인지 인식하는데 어려움이 생긴다.

- '삼'의 경우, '삼' '쌈' '사' '싸' 와 같이 발음된다. 이는 '삼'의 위치에 따라, 앞뒤에 오는 숫자음의 경우에 따라 발음되는 경우를 모두 포함한 것이다.

한국어 숫자음의 경우 비슷한 발음으로 인해 인식오류를 나타내는 단어에 대한 인식실험을 진행하였고, Discriminant Learning을 통해 인식률 향상을 확인하였다.

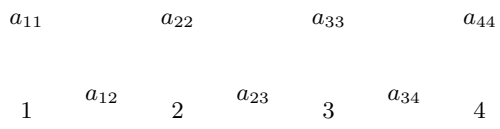
실험결과 이러한 HMM학습자료를 기반으로 한 음성인식 시스템은 '일'에 대한 다양한 발음에 대해 매우 높은 인식률을 가짐을 확인하였다. 그러나 '일'에 대해 너무 포괄적인 학습이 이루어져 '칠'에 대한 인식률 저하를 초래하였다. 즉, '칠'에 대한 실험데이터들이 '일'로 인식되는 결과를 낳았다. 이러한 결과에 대한 개선책으로 Discriminant 학습을 적용하였고, '칠'에 대한 인식률의 경우 괄목할만한 향상을 가져왔다.

따라서, 다양하게 발음되는 숫자음이라도 HMM과 Discriminant 학습을 이용해 하나의 학습자료로 만들고, beam-search 등의 방법을 쓰면 충분히 높은 인식률을 가진 음성인식 시스템을 구축할 수 있다.

제 3 절 HMM을 활용한 음성인식

HMM을 기반으로 한 음성인식 시스템은 그 탁월한 인식률이 이미 검증되었다. 일반적으로 HMM은 N개의 상태 수를 가진 left-to-right 모델로 구성되며, 상태별 초기확률을 나타내는 $\Pi = \{\pi_i : 1 \le i \le N\}$, 상태 전이 확률 행렬 $A = \{a_{ij} : 1 \le i, j \le N\}$, 상태별 관찰 확률을 나타내는 $B = \{b_i : 1 \le i \le N\}$을 구성 원소로 하는 벡터 $\lambda = (\Pi, A, B)$로 모델을 표현한다.

본 논문에서의 HMM 모델은 아래의 그림과 같이 개별 숫자음 데이터에 대해 3개 또는 4개의 상태 수를 가지고, 상태별 전이는 2개 만 가능한 left-to-right 모델로 하였다.



행렬 A는 다음과 같이 임의로 만들어 시작한다.

$$\begin{pmatrix} 0.55 & 0.45 & 0 & 0 \\ 0 & 0.55 & 0.45 & 0 \\ 0 & 0 & 0.55 & 0.45 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

이 행렬은 viterbi 알고리듬을 이용하여 최적의 상태변화를 나타내는 행렬로 수렴한다. 이 과정을 살펴보면

(1) 모든 입력벡터들을 각 생태에 같은 수로 배정한다.

(2) 각 상태에 모인 벡터들에 대해 K-Means 알고리듬을 이용하여 두 개의 cluster로 나누고, 각 cluster에 있는 음성벡터들의 평균벡터 μ_{ij}와 분산벡터 σ_{ij}를 구한다. 여기에서

$\mu_{ij} = (\mu_{ij}^1, \mu_{ij}^2, \cdots, \mu_{ij}^{26})$, $1 \le j \le 5, i = 1, 2$

: 상태 q_j의 cluster i에 에 있는 벡터들의 평균벡터

$\sigma_{ij} = (\sigma_{ij}^1, \sigma_{ij}^2, \cdots, \sigma_{ij}^{26})$, $1 \le j \le 5, i = 1, 2$

: 상태 q_j의 cluster i에 에 있는 벡터들의 분산벡터

와 같은 26차원 벡터를 나타낸다.

(3) 상태 q_j에서 관측벡터 x_k를 관찰하게될 확률 b_{jk}를 구한다. b_{jk}는 연속 확률밀도 함수를 사용하여

$$b_{jk} = \sum_{i=1}^{2} C_{ji} \frac{1}{\sqrt{2\pi}^{26} \sigma_{ji}^1 \sigma_{ji}^2 \cdots \sigma_{ji}^{26}} \exp\left(-\frac{1}{2} \prod_{s=1}^{26} \frac{(x_k^s - \mu_{ji}^s)^2}{\sigma_{ji}^s} \right)$$

이와 같은 과정을 반복함으로써 주어진 음성 벡터 $X = (x_1, x_2, \ldots, x_{T_f})$에 대해 Viterbi 알고리즘에 의한 최적 상태변화가 $\mathbf{q} = (q_0, q_1, \ldots, q_{T_f})$라면 maximum likelihood 값은

$$\mathbb{L}(X) = \sum_{\boldsymbol{q}} \pi_{q_0} \prod_{t=1}^{T_f} a_{q_{t-1} q_t} b_{q_t}(x_t)$$

로 계산한다. 여기에서 $b_{q_t}(x_t)$는 상태 j에서 x_t를 관찰하게 될 확률을 나타내며, 상태별로 모인 벡터들을 clustering한 후, cluster 별로 평균과 분산을 이용한 정규분포로 계산된다. HMM은 보통 maximum likelihood 값을 계산 함으로써 음성 패턴을 찾아가는 통계적인 모델로 특징 지울 수 있다. 그러나 HMM은 많은 학습자료를 필요로 한다는 단점을 지닌다.

제 4 절 HMM을 기반한 Discriminant 학습

Discriminant 학습은 비슷한 발음 등으로 인식오류의 가능성이 있는 다른 음성모델을 함께 고려하는 학습시스템으로서 MCE(Minimum Classification Error)를 기반으로 한다. MCE를 최소로 하는 방향으로 HMM 학습 결과를 개선시키는 Discriminant 학습으로

(1) 화자확인, 화자인식등에 광범위하게 적용되고 있으며,

(2) 불충분한 학습자료를 사용한 HMM학습결과로 초래되는 인식률 저하를 개선

을 기대할 수 있다.

K개의 HMM학습 자료를 $\Lambda = \{\lambda_k, 1 \leq K \leq\}$로 표시하고, 각각의 음성 패턴 X_i $(1 \leq i \leq X_T)$는 K개의 음성 클래스 C_k, $(1 \leq k \leq K)$ 중 하나에 포함된다. 여기에서

X_T는 입력 데이터의 수를 나타낸다. HMM에 기반한 Discriminant 학습은 Viterbi 알고리즘에 의한 log likelihood의 값을 Discriminant 함수로 사용한다. HMM 학습의 개선은 GPD(Generalized Probabilistic Descent) 방법을 사용하여 반복적으로 MCE를 최소화시키는 학습결과를 만들어낸다.

HMM에 기반한 GPD는 다음과 같은 세 가지 함수를 구성요소로 한다.

(a) **Discriminant 함수** $g_k(X_i;\Lambda)$: 음성 패턴 X_i가 클래스 C_k에 대한 HMM likelihood 값

$$g_k(X_i;\Lambda) = \log\left[\mathbb{L}_k(X_i)\right]$$

을 Discriminant 함수로 사용한다. 따라서 음성 패턴X_i는 Viterbi 알고리즘에 의한 likelihood 값이 최대인 클래스 C_k에 속하는 것으로 인식된다.

$$X_i \in C_j \quad \text{if} \quad g_j(X_i;\Lambda) = \max_k g_k(X_i;\Lambda)$$

(b) **Misclassification Measure** $d_k(X_i;\Lambda)$:

$$d_k(X_i;\Lambda) = -g_k(X_i;\Lambda) + G_k(X_i;\Lambda)$$

여기에서 $G_k(X_i;\Lambda)$는 클래스 C_k에 들어가야 할 입력데이터 X_i가 다른 클래스에 들어가는 모든 경우를 의미하며 다음과 같이 다른 클래스들의 likelihood 값들의 기하평균의 log값으로 계산한다.

$$G_k(X_i;\Lambda) = \log\left(\frac{1}{K-1}\sum_{i,j\neq k}\exp\left[\eta g_j(X_i;\Lambda)\right]\right)^{1/\eta}.$$

일반적으로 $\eta = 1$로 놓는다. $d_k(X_i;\Lambda)$의 정의로부터 입력 데이터 X_k가 올바르게 인식되었다면 d_k의 값이 음의 값을, 잘못 인식된 경우는 양의 값을 가짐을 알 수 있다.

(c) **Loss 함수** $l_k(X_i;\Lambda)$: GPD 방법은 미분을 이용해 파라미터 Λ를 반복적으로 보정해가는 방법이다. 따라서 Loss 함수는 다음과 같이 0-1사이의 값을 갖는 sigmoid 함수로 정의된다. 이 sigmoid 함수는 단조증가이며 미분가능한 함수이다.

$$l_k(X_i;\Lambda) = \ell(d_k) = \frac{1}{1+\exp[-a(d_k+b)]}$$

여기에서 $a(>0)$는 상수이고, d_k+b 부근에서 sigmoid 함수의 기울기를 나타낸다. 또한 적당한 b값의 설정은 sigmoid 함수의 미분 값이 0을 갖게 되는 것을 방지해 파라미터 Λ의 변화를 가능하게 한다. 일반적으로, GPD에 의한 파라미터 Λ의 변화는 misclassification measure d_k의 값이 $-b$의 근처에 있을 때, 결정적으로 일어나게 된다. 따라서 효율적인 GPD 알고리즘을 구현하기 위해서는

$$b = d_{\min} = -\max_{1\le i\le X_T, 1\le k\le K} [-d_k(X_i;\Lambda)]\,\chi(X_i\in C_l)$$

로 설정하는 것이 좋다. 이 때, χ는 다음과 같이 정의된다.

$$\chi(W)=\begin{cases}1, & W\text{가 참일 때,}\\ 0, & W\text{가 거짓일 때.}\end{cases}$$

이와 같은 세 가지 구성요소를 가진 GPD방법에 의해

$$\Lambda_{t+1}=\Lambda_t-\epsilon_t\nabla\ell(X_t,\Lambda)\mid_{\Lambda=\Lambda_t}$$

를 따라 수렴하는 파라미터 Λ를 구하는 것으로서 결국, GPD에 의한 Discriminant 학습은 다음과 같은 평균오차율(everage error rate)을 최소로 하는 파라미터 Λ를 찾아가는 방법이다.

$$L(\Lambda)=\frac{1}{N}\sum_{i=1}^{X_T}\sum_{k=1}^{K} l_k(X_i;\Lambda)\chi(X_i\in C_k).$$

여기에서 ϵ_t은 매우 작은 수로서 이 논문의 실험에서는 $\epsilon_t=1-t/75$를 사용하였다.

클래스 C_j, $(1\le j\le K)$의 HMM 파라미터에 대한 미분 $\nabla l_k(X;\lambda)$는

$$\nabla_{\Lambda_j} l_k(X;\lambda)=\frac{\partial\ell_k}{\partial d_k}\frac{\partial d_k}{\partial g_j}\cdot\nabla_{\Lambda_j}[g_j(X;\Lambda)]$$

로 유도되며, 각각의 편미분은

$$\frac{\partial\ell_k}{\partial d_k}=a\cdot\ell_k(1-\ell_k), \tag{1}$$

$$\frac{\partial d_k}{\partial g_j}=\begin{cases}-1 & j=k\text{일 때,}\\ \frac{\exp[\eta g_j(X,\Lambda)}{\sum_{n,n\neq k}\exp[\eta g_n(X,\Lambda)]} & j\neq k\text{일 때}\end{cases}$$

이다. HMM 파라미터에 대한 $\nabla_{\Lambda_j}[g_j(X;\Lambda)]$는 클래스 j에서 전이확률 행렬의 원소 $a_{q_t,q_{t+1}}$, $1 \le t \le N$에 대한 편미분과 상태별 관찰 확률 $b_q(X)$의 계산에 필요한 평균(μ), 분산(σ), cluster 계수 c대한 편미분으로 나타난다 [2], [5].

식 (1)에서 d_k가 b에서 조금만 떨어져 있어도 $\frac{\partial \ell_k}{\partial d_k}$의 값은 exp 함수의 특성으로 인해 0에 가까운 값을 갖게된다. 따라서 b값을 설정하게 한 입력 데이터 X와 극히 제한된 수의 입력 데이터를 제외하고는 GPD에 의한 파라미터 보정에 영향을 주지 못한다.

제 5 절 실험 및 결과

한국어 연속 숫자음 음성인식 실험을 위해 1000여명의 화자가 '일', '0|', ⋯, '구', '영', '공'의 숫자음으로 이루어진 연속 숫자음을 전화를 통해 발음하게 하여 학습 데이터로 사용하였다. 인식 단위는 다음과 같이

일, 릴, 길, 밀, 0|, 리, 기, 미, 삼, 쌈, 사, 싸, 오, 로,
고, 모, 육, 룩, 유, 류, 칠, 치, 파, 팔, 구, 공, 꽁, 영, 녕

모든 숫자음 발음과 그 숫자음이 발음 위치, 음운현상에 따라 변화된 발음을 모두 인식단위로 설정하였다. 인식 실험을 위한 데이터는 같은 숫자음을 100인이 전화로 발음한 것을 사용하였다. 음성 인식에 필요한 feature는 15msec의 구간으로 잘라 12-차원 cepstrum과 power, 12-차원 delta-cepstrum과 power의 26 차원 벡터로 추출하였 다.

각 숫자음에 대해서 '일', '삼'과 같이 받침이 있는 숫자음은 4개의 상태를 가진 모델로, '0|', '오'와 같이 받침이 없는 숫자음은 3개의 상태를 가진 모델로 설정하였다. k-means 알고리즘[3]을 이용하여 상태별로 cluster의 수는 모두 3개로 하였으며, HMM 모델은 연속형 left-to-right 모델로 설정하였다. covariance matrix로는 diagonal형을 채용하였다.

실험결과 HMM 알고리즘의 효율성이 충분히 드러나 고립 숫자음의 경우 모두 95%이상의 높은 인식률을 나타내었다. 그러나, 연속 숫자음의 경우에는 '칠'의 인식률이 매우 저조하였다. 발음이 비슷한 '일', '0|', '칠'세 숫자음에 국한한 인식률은 다음 표1과 같다.

숫자음	'일'	'0|'	'칠'
실험 데이터의 수	56	56	56
인식률	100%	100%	78.57%

[표1] : HMM을 활용한 숫자음 인식률.

표1에서와 같이 '칠'에 대한 인식률이 저조한 것은 '칠'의 발음이 '일'로 인식되고 있기 때문이다. 이는 '일'에 대한 HMM 학습이 '일', '밀', '길', '밀'을 포함하도록 매우 폭넓게 이루어졌기 때문이라 생각된다.

'칠'에 대한 낮은 인식률을 높이기 위해 Discriminant 학습을 이용하였다. Discriminant 학습에 사용한 데이터는 '일'로 인식되는 '칠'의 발음 가운데 d_k의 값이 작고(이는 두 발음이 비슷하여 Maximum likelihood 값의 차이가 작게 나타남을 의미), 모든 '칠'에 대한 음성이 갖는 Maximum likelihood 값의 평균에 가까운 것을 2개 정도로 충분하였다. Discriminant 학습을 이용한 HMM 파라미터의 보정을 사용하여 다시 인식률을 실험한 결과는 다음 표2와 같다.

숫자음	'일'	'0|'	'칠'
실험 데이터의 수	56	56	56
인식률	100%	100%	98.21%

[표2] : Discriminant 학습을 활용한 숫자음 인식률.

실험 결과 '칠'에 대한 인식률은 약 20%향상된 것으로 나타났다.

연속 숫자음의 인식은 빔서치 알고리즘을 이용하였다. 이는 시간의 변화에 따라 HMM 상태의 전이 뿐 아니라 각각의 시간에서 다른 숫자음으로의 전이를 함께 고려하는 방법으로 음성인식에 있어 가장 널리 쓰이는 방법이다. 두 자리 숫자음에 대한 인식은 첫 번째 숫자음의 인식에 이어 두 번째 숫자음의 인식을 연결하여 Maximum likelihood 의 값이 가장 결과를 출력함으로써 가능하다.

숫자음 '일'이 뒤에 발음되는 경우, 앞에 발음된 숫자음에 따라 '밀', '릴', '길', '일' 등의 네 가지로 발음이 되며, 이것은 '미', '리', '기', '0|' 등의 네 가지로 발음되는 '0|'와 인식의 혼란을 가져올 것으로 예상하였다. 그러나, 실제 인식에 있어서는 일의 다양한 발음이 오히려 '칠'과 혼란을 가져왔다. 이로 인한 인식률의 저하는 Discriminant 학습으로 충분히 해결하였다. 또한 '삼'과 사'의 경우는 고립 숫자음에 대해서는 HMM 학습으로 충분히 높은 인식률을 나타내지만 두 숫자음이 두 자리 연속 숫자음의 앞 음절에 발음된 경우에 인식의 혼란을 가져온다. 예를 들어 35와 45의 경우는 각각 '삼오'

와 '사오'로 발음되어 인식에 있어 문제가 없지만 32 와 42 의 경우는 '사미와 '사이'로 발음되어 인식에 있어 혼란을 가져온다. 이에 대한 해결책으로도 Discriminant 학습이 제시될 수 있으며 실험결과 인식률의 향상을 가져왔다. 31, 32, 35 등과 같이 '삼'의 발음이 '사'와 비슷하게 발음되는 실험자료에 대해 숫자음 '삼'에 대해 다음과 같은 인식률 향상을 보였다.

'삼'	HMM 학습결과	Discriminant 학습결과
31	78.4%	86.8%
32	80.2%	85.7%
35	71.2%	81.8%

[표3] : 연속 숫자음 '삼'의 인식률 향상

이 실험을 통해 HMM 학습의 우수한 인식률을 확인하였으며, HMM 학습으로 인식률이 낮은 경우, 즉 비슷한 발음으로 인해 인식률의 저하가 나타나거나, 포괄적인 학습으로 인하여 인식률이 낮은 경우에 Discriminant 학습으로 매우 높은 인식률의 향상을 가져올 수 있었다.

또한, HMM 학습을 위해서는 충분한 양의 학습 데이터가 필요한데, 위의 고립 숫자음의 경우 적은 수의 데이터를 사용했을 경우, 인식률이 낮다. 이에 대한 해결책으로 Discriminant 학습이 적용될 수 있으며, 실험을 통해 이 경우에도 높은 인식률 향상을 가져왔다.

참고 문헌

[1] Jiqing Han, Munsung Han, Byu-Bong Park, Jeongue Park and Wen Gao, "Discriminative learning of additive noise and channel distortions for robust speech recognition", in *Proc. ICASSP-98* pp. 81–84.

[2] Biing-Hwang Juang, Wu Chou and Chin-Hui Lee, "Minimum classification error rate methods for speech recognition", *IEEE Trans. Speech and Audio Process.*, textbf5 (3), May, pp. 257–265, (1997).

[3] Biing-Hwang Juang and L. Rabiner, "The segmental K-means algorithm for estimating parameters of hidden Markov models", *IEEE Trans. Audio Speech Signal Process.*, **38**, pp. 1639–1641, (1990).

[4] Chin-Hui Lee and Lawrence R. Rabiner, "A frame-synchronous Network Search Algorithm for connected word recognition", *IEEE Trans. Acoust. Speech Signal Processing*, **37**, Nov., pp. 1649–1658, (1989).

[5] Chi-Shi Liu, Chin-Hui Lee, Wu Chou, Biing-Hwang Juang and Aaron E. Rosenberg, "A study on minimum error discriminative training for speaker recognition", *J. Acoust. Soc Am.* **97** (1), January, pp. 637– 648, (1995).

한국전자통신연구원(ETRI)