

대용량 연속음성 인식을 위한 효율적인 탐색 알고리즘

박형민*, 최인정*, 이수영*, 김도영**, 김상용**

*한국과학기술원 전기 및 전자공학과, **삼성종합기술원 휴먼 인터페이스 연구실

Efficient Search Algorithms for Continuous Speech Recognition

Hyung-Min Park*, In-Jeong Choi*, Soo-Young Lee*,
Do-Yeong Kim** and Sang-Yong Kim**

Dept. of Electrical Eng., Korea Advanced Institute of Science and Technology

**Human Interface Lab., Samsung Advanced Institute of Technology

E-mail : mhp@neuron.kaist.ac.kr

요 약

이 논문에서는 대용량 연속음성 인식에서 인식 속도를 향상시키기 위한 방법들에 대해서 연구하였다. 음성인식에 있어서 많은 양의 계산을 요하는 부분은 관측 확률의 계산과 탐색에 필요한 계산이다. 탐색에 필요한 계산을 줄이기 위하여 빔 탐색법과 phoneme look-ahead 기법을 통해 탐색 공간을 줄였으며, 관측 확률을 계산하는데 소요되는 시간을 줄이기 위하여 입력 특징 벡터와 이웃 관계에 있는 가우시안 성분들만 정확한 계산을 하는 VQ에 의한 계산량 감축 방법과 tree-structured pdf 방법을 구현하였다. 3천개의 어휘와 2천여개의 트라이폰 모델로 구성된 연속 음성인식 시스템에서 보통의 Viterbi 빔 탐색법을 적용한 경우에 실시간의 2.73배의 인식 속도로 93.39%의 단어 인식률을 얻을 수 있는데 phoneme look-ahead 기법과 tree-structured pdf 방법을 추가 적용함으로써 비슷한 인식 성능에서 1.55배의 인식 속도를 얻을 수 있었다.

1. 서론

음성인식 시스템이 광범위한 분야에서 활용되기 위해서는 대용량 어휘와 자연스러운 형태로 발음된 음성을 처리할 수 있어야 한다. 그러나, 대용량 연속음성 인식은 고립단어 인식에 비해 여러 가지 이유 때문에 어려운 것이 일반적이다[1][2]. 연속어 인식을 위해서는 모든 가능한 단어 경계를 고려하여 가장 가능성이 큰 단어 열을 찾아야 하며, 이를 위해서는 효율적인 탐색 방법과 상세한 언어 모델의 사용이 필수적이다. 또한 자연스러운 회화체 음성에서 나타나는 조음화 결함을 반영하는 상세한 음향학적 모델이 필요하다.

음성인식의 계산 비용은 크게 두 가지 성분으로 분류할 수 있다[3]. 하나는 관측 확률의 계산 비용이며, 나머지는 탐색에 필요한 계산 비용이다. HMM (hidden Markov model)을 사용하는 시스템의 경우에서, 관측 확률의 비용은 모든 HMM state에서 주어진 특징 벡터를 관측 확률을 계산하는데 필요한 비용이며, 특히 연속 혼합모델도 HMM과 복잡한 음향 모델을 사용할 경우 이 비용이 전체 계산 비용에서 50-80%를 차지한다. 탐색 비

용이란 완전한 음성 입력이 주어졌을 때 가장 가능성 있는 단어열을 탐색하는데 필요한 계산량으로 어휘의 수가 많아질수록 탐색 비용은 증가하며, 대용량 연속어 인식에서 탐색 비용의 중요성이 증대된다. 본 논문에서는 두 비용의 감소를 위한 방법들에 대해 연구하였다.

2. Viterbi 빔 탐색과 활성 탐색 영역 리스트를 이용한 인식 속도 개선

Viterbi 탐색 알고리즘은 One-Pass 탐색 알고리즘이라고도 불리며, 동적 프로그래밍 알고리즘에 기반을 두고 있다. 탐색 공간은 이차원으로 형성되며, 한 차원은 관측 열, 그리고 다른 차원은 모든 단어 모델을 표현하는 HMM state의 열로 이루어진다. Viterbi 탐색 알고리즘은 time-synchronous하게 진행되며, 첫번째 프레임부터 시작하여 관측 열과 HMM state 열의 2차원 공간에서 가장 가능성 높은 경로, 즉 단어 열을 탐색한다.

대용량의 어휘인 경우, Viterbi 탐색 알고리즘에서 모든 state들을 탐색하는 것은 너무나 많은 계산량과 긴 탐색 시간을 초래한다. 빔 탐색에서는 매 프레임에서 모든 후보 경로들을 계산하지 않고 가능성이 높은 후보들만을 계산한다. 가능성이 희박한 가설들을 절단하기 위해 사용되는 방법들에는 표준 빔 pruning과 히스토그램 pruning 등이 있다. 표준 빔 pruning 방법은 먼저 활성 state들 중에서 최대 likelihood 값을 찾고, 최대값보다 임계치 이하인 likelihood 값을 가지는 state는 고려하지 않는다.

히스토그램 pruning 방법은 매 프레임에서 활성 state 가설들의 수를 미리 정해진 최대 수로 제한하는 방법이다[4]. 이것은 고속 정렬 알고리즘이나 히스토그램 분포를 이용하여 수행되어진다. 고정된 빔 사이즈를 사용할 경우, 매 프레임마다 likelihood 값의 차수가 다르기 때문에 낮은 값의 프레임에서는 활성 state의 수가 많고, 높은 값의 프레임에서는 활성 state의 수가 적어 탐색 에러를 발생할 가능성이 높다. 따라서 적당히 큰 빔을 사용해야 하며, 낮은 값의 프레임에서는 활성 state 가설들의 수를 제한할 필요가 있는 것이다.

빔 탐색에서 활성 state를 효과적으로 관리하기 위하

어 활성 탐색 영역의 리스트를 구성한다[5]. 그림 1(a)는 빔 탐색에서의 경로 결합에 대해 설명하고 있으며, 그림 1(b)는 이러한 활성 탐색 영역의 리스트 구성에 대하여 설명하고 있다. 그림 1(a)에서 검게 칠해진 부분은 프레임 $t-1$ 에서의 활성 state들을 나타내며, 굵은 원의 노드들은 HMM 모델의 state 전이 구조에 따라 프레임 t 에서 동적 프로그래밍 알고리즘이 수행되어질 state 가설들을 나타낸다. 그림 1(b)에서 $k(l)$ 은 l 번째 순서의 단어 인덱스를, 그리고 $b(l)$ 과 $e(l)$ 은 해당 활성 영역의 시작과 마지막 state 가설에 대한 포인터를 나타낸다. 이와 같은 방법으로 모든 state 가설들을 검토하는 요구를 제거하고 경로 결합 규칙을 효율적으로 적용하여 수행되어질 계산량을 크게 개선할 수 있다.

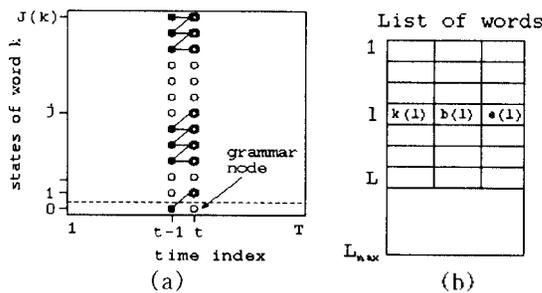


그림 1. Viterbi 빔 탐색
(a) 경로 결합 (b) 활성 탐색 영역의 리스트 구성

3. Phoneme Look-ahead (PLA) [6]

매 프레임마다 탐색 과정에서 새롭게 시작될 음소 모델의 가설이 생기며, 먼저 이 새로운 가설이 다음 미래의 프레임 동안에 수행될 pruning 단계에서 살아남을 가능성이 있는지를 먼저 검토해 보는 것이 PLA (phoneme look-ahead) 방법이다.

PLA pruning을 위하여 주어진 프레임 t 에서, 새롭게 활성화 될 음소 모델 중의 하나 α 에 대하여 다음 스코어를 계산한다.

$$\hat{Q}_v(t, \alpha) = \hat{q}(\alpha; t, \Delta t) \cdot Q_v(t, S_\alpha) \quad (1)$$

여기서 $\hat{q}(\alpha; t, \Delta t)$ 는 음소 모델 α 가 특징 벡터들 $x_{t+\Delta t}, \dots, x_{t+\Delta t}$ 를 발생시킬 확률을 나타내며, Δt 는 음소 지속시간의 평균 정도의 값이 사용된다. 또, $Q_v(t, s)$ 는 프레임 t 와 state s 의 노드 (t, s) 에서 끝나는 가장 가능성 높은 부분 경로의 확률을 나타내며, v 는 state s 가 속한 단어 인덱스를 나타낸다. 그리고, $\tilde{\alpha}$ 는 단어 발음사전에서 α 의 유일한 바로 이전 모델을 나타내며, $S_{\tilde{\alpha}}$ 는 음소 모델 $\tilde{\alpha}$ 의 마지막 state를 나타낸다. 빔 pruning 방법에서와 같이, pruning 과정은 프레임 t 에서 고려되는 모든 가설들의 최고 스코어 $Q_{LA}(t)$ 에 근거하여 이루어지며, 실험적인 확인을 통해서 성능면에서의 손실없이 다음과 같은 방법에 의해 $Q_{LA}(t)$ 의 값을 근사화할 수 있다.

$$Q_{LA}(t) \equiv \max_{\alpha} \{\hat{q}(\alpha; t, \Delta t)\} \cdot \max_{(v, \beta)} \{Q_v(t, S_{\beta})\} \quad (2)$$

여기서 β 는 임의의 음소 모델을 나타낸다.

Look-ahead 스코어 $\hat{q}(\alpha; t, \Delta t)$ 를 계산하기 위하여, 음소 모델과 특징 벡터열 $x_{t+\Delta t}, \dots, x_{t+\Delta t}$ 과의 정합을 수행한다. 이것을 위해 $\phi_{\alpha}(r, s; t)$ 는 특징 벡터열 $x_{t+\Delta t}, \dots, x_{t+\Delta t}$ 가 음소 모델 α 의 states $1, \dots, s$ 와 정합하는 스코어를 정의한다고 하자. 3개의 state를 갖는 HMM에 대하여, 그림 2는 look-ahead 스코어를 계산하는 방법을 설명하고 있다. 그림에서 빔 검진 부분들은 look-ahead 정렬 과정이 완료될 수 있는 가능한 state들을 표시한다. Look-ahead 스코어를 계산하기 위하여 빔 영역에 속한 state들의 스코어를 고려하여야 하며, 또한 다른 지속시간 τ 의 영향이 반영되도록 스코어를 지속시간에 대해 정규화하여야 한다. 따라서 look-ahead 스코어는 다음 식에 의해 계산되어진다.

$$\hat{q}(\alpha; t, \Delta t) = \max \{ \max_s \{ \phi_{\alpha}(\Delta t, s; t) \}, \max_{\tau} \{ \phi_{\alpha}(\tau, S; t)^{\Delta t/\tau} \} \} \quad (3)$$

여기서 S 는 HMM의 마지막 state를 나타낸다.

탐색 시간을 개선하기 위해서는 음소 확장에 의한 look-ahead 스코어의 계산에 소요되는 계산 비용이 충분히 작아야 하므로 간단한 음소 모델을 사용하고, 두 프레임 정도마다 PLA 과정을 수행한다.

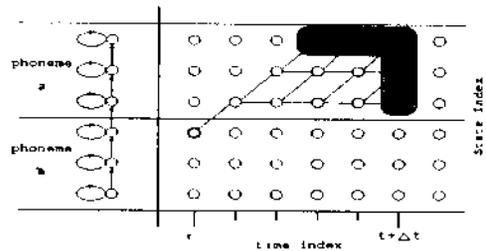


그림 2. 3-state HMM을 이용한 PLA

4. 연속 혼합말도 HMM에서 관측 확률의 계산량 감축

가우시안 모델들은 입력 특징 벡터가 가우시안 평균에 가까이 있는 경우에는 통계적으로 정확하지만 그 분포의 꼬리 부분에 위치할 때에는 단지 근사화된 likelihood를 제공한다. VQ를 이용한 계산량 감축 방법은 입력 벡터와 neighborhood의 관계에 있는 가우시안 성분들에 대해서만 정확한 likelihood를 계산하며, 가우시안 분포의 꼬리 부분에 위치하게 되는 가우시안들의 likelihood는 table look-up이나 작은 상수 값에 의해 근사화 된다[7]. 시스템의 학습 동안에 모든 가우시안 성분들은 neighborhood들로 군집화 되는데 가우시안들의 각 neighborhood마다 하나의 코드워드로 구성되도록 벡터 양자화기를 정의한다. i 번째와 j 번째의 가우시안 성분간의 거리 척도로서 다음 식을 사용한다.

$$\delta(\mu_i, \mu_j) = \frac{1}{p} \sum_{k=1}^p \{ \omega(k) (\mu_i(k) - \mu_j(k)) \}^2 \quad (4)$$

여기서 $\omega(k)$ 는 평균 공분산 행렬의 k 번째 대각 성분의 제곱근의 역과 같다. VQ에 의한 가우시안 성분들의 선택적인 likelihood 계산 방법이 인식 성능의 큰 저하

를 초래하지 않기 위해서는 더 나은 neighborhood에 대한 정의가 필요하므로 다음 식을 사용한다.

$$N(\cdot, \mu_m, U_m) \in v_\phi \quad (5)$$

$$\text{iff } \frac{1}{p} \sum_{k=1}^p \frac{(c_\phi(i) - \mu_m(i))^2}{U_{avg}(i)} \leq \Theta$$

즉, 코드워드 c_ϕ 의 neighborhood v_ϕ 는 위의 조건식을 만족하는 가우시안 성분들로 구성된다. 위 식에서 Θ 는 neighborhood의 정의에 사용되는 임계치($\Theta \gg 1$)이며, $U_{avg}(i)$ 는 평균 공분산 행렬의 i 번째 대각 성분을 나타낸다.

또 한가지 방법은 더욱 정확히 계산해야 할 가우시안 성분들과 그렇지 않은 가우시안 성분들을 효율적으로 분류하여 계산하기 위하여 트리 구조의 확률 밀도 함수를 이용한다[8]. 트리 구조의 확률 밀도 함수에서, 각 가우시안 성분은 트리의 노드로서 구조화된다. 상위 레벨의 트리 노드에서의 가우시안은 다음 레벨 아들 노드의 가우시안 확률 밀도 함수들의 집합에 해당되기 때문에 cluster 확률 밀도 함수라고 불린다. Cluster 확률 밀도 함수는 하나의 가우시안 확률 밀도 함수를 사용하여, 그 cluster의 모든 아들 확률 밀도 함수들의 성분을 근사화함으로써 얻어진다. 그림 3은 트리로 구성된 확률밀도 함수의 구조를 설명하고 있다.

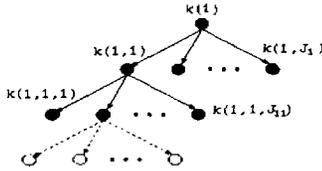


그림 3. 트리 구조의 확률 밀도 함수의 구조

인식 동안에 상위 레벨의 노드들이 정해진 선택 기준에 따라 먼저 선택되고 트리의 모든 레벨을 따라 아래로 진행된다. 일반적으로 사용되고 있는 선택 기준중의 하나는 N-best 선택 규칙으로 cluster 확률 밀도 함수의 likelihood가 그 레벨에서 상위 N개에 포함되면 활성 노드로 표시된다. 각 레벨에서 부모 노드가 선택된 노드들의 likelihood가 계산되며, likelihood에 근거하여 선택 노드들이 다시 결정된다. 그림 3에서 점선에 의해 연결된 base 노드, 즉 element 확률 밀도 함수들은 바로 위의 부모 노드가 선택된 경우에만 likelihood의 계산이 수행되고, 그렇지 않은 경우에는 가장 가까운 선택된 부모 노드에서의 likelihood 값으로 근사화된다. 일반적으로 트리 레벨의 수는 3이하이며, 각 레벨에서의 노드 수는 재귀된 법칙을 따른다.

트리 구조의 확률 밀도 함수를 설계할 때 divergence [9]라는 거리 척도를 사용한다. 대각 공분산 행렬에서 두 개의 가우시안 확률 밀도 성분 f_k 와 f_m 사이의 divergence 척도는 다음 식과 같다.

$$\delta(f_k, f_m)$$

$$= \sum_i \left(\frac{\sigma_k^2(i) + \Delta_{km}^2(i)}{2\sigma_m^2(i)} + \frac{\sigma_m^2(i) + \Delta_{km}^2(i)}{2\sigma_k^2(i)} - 1 \right) \quad (6)$$

where, $\Delta_{km}^2(i) = (\mu_k(i) - \mu_m(i))^2$

군집화 알고리즘은 하나의 cluster에 속한 가우시안 확률 밀도 함수들을 cluster 가우시안 확률 밀도 함수로 근사화한다. Cluster 확률 밀도 함수 f_m 의 평균과 분산은 다음 식들에 의해 추정되어진다.

$$\mu_m(i) = \frac{1}{K} \sum_{k=1}^K \mu_k(i) \quad (7)$$

$$\sigma_m^2(i) = \frac{1}{K} \left[\sum_{k=1}^K \sigma_k^2(i) + \sum_{k=1}^K \mu_k^2(i) - K\mu_m^2(i) \right] \quad (8)$$

5. 실험 결과 및 고찰

음성 인식 시스템의 학습과 성능 평가를 위하여 사용된 음성 데이터베이스는 한국과학기술원 통신연구실에서 제작한 무역 상담용 연속음성 데이터베이스[10]이다. 문장은 2,920개의 어휘로 구성되어 있으며, 녹음은 조용한 사무실 환경에서 이루어졌으며, 16kHz, 16bit로 표본화 되었다. 한 문장 당 평균 단어 수는 8.4개이며, 발음속도는 평균 분당 166.5개의 단어를 발음하였다. 학습에는 남성화자 75명의 데이터를, 성능 평가에는 학습에 참가하지 않은 남성 25명이 발음한 2,421개의 문장이 사용되었다.

음성 데이터는 10msec씩 매 프레임마다 이동하면서 30msec 길이의 샘플에 대해 특징 벡터를 추출하였다. 사용된 특징 벡터는 12차의 MFCC와 에너지 및 그들의 1차 미분 계수이다. 각 문장마다 mean subtraction 과정을 거쳐 최종적인 MFCC를 얻었으며, delta-MFCC 계수를 구하기 위해 앞뒤 2프레임에 대한 벡터들의 변화량을 고려하였다. 음성을 표현하기 위한 모델로서 음소와 트라이폰을 사용하였다. 이중 모음을 고려한 기본 음소 모델의 수는 41개이며, 트라이폰 모델은 2,045개이다. 인식 실험에서는 음성을 표현하는 단위로서 트라이폰 모델을 사용하였으며, 음소 모델은 PLA pruning에서 필요한 정보를 얻기 위해 추가적으로 사용되었다. 언어모델로는 단어 클래스에 기반한 bigram을 적용하였다.

표 1. 빔 사이즈에 따른 인식 성과와 탐색 속도

빔 사이즈	단어 인식률	문장 인식률	N_{avg}	$xRealTime$
80	90.30	67.08	1,436	0.89
100	92.53	69.27	2,225	1.26
120	93.17	69.89	3,240	1.76
150	93.39	70.26	5,141	2.73

표 1은 Viterbi 표준 빔 pruning에서 빔 사이즈에 따른 인식 성과와 탐색 속도를 나타내고 있다. 표에서 N_{avg} 는 탐색 과정에서 매 프레임당 활성 state의 평균 수를 나타내며, 탐색 공간의 크기를 나타낸다. 참고로 전체 HMM state의 총 수는 60,600여개이다. 또한 $xRealTime$ 은 음성신호의 길이에 대한 인식 시간의 비를 나타내며, 탐색 속도의 척도로 사용되었다. 시스템의 성능 평가(탐색 속도)는 Digital사의 Alpha 스테이션 컴퓨터를 사용하여 측정되었으며, CPU의 처리 속도는 333MHz이다. 일반적으로 알려진 사실과 같이, 작은 빔 사이즈를 사용할 경우 활성 탐색 영역을 줄여 탐색 시간을 크게 개선할 수 있지만 인식 성능의 악화를 감수

하여야 한다. 또, 결과에서 120과 150의 빔 사이즈가 사용될 때, 탐색 시간이 55% 증가한 반면 단어 오인식률은 3% 정도의 감소에 그쳐 어느 정도 인식 성능이 포화됨을 보여준다.

다음은 히스토그램 pruning을 적용하여 활성 노드의 수가 정해진 최대수보다 많을 때, 누적 likelihood에 근거하여 초과된 만큼의 노드 수를 pruning하였다. 히스토그램 분포를 사용하여 많은 시간이 소요되는 정렬 과정 없이 누적 분포에 따라 가능성이 떨어지는 노드들을 pruning 할 수 있다. 실험 결과는 표 2에 나타나 있으며, 이 실험에서는 150의 빔 사이즈를 사용하였다. 히스토그램 pruning을 적용하였을 때 탐색 속도는 24%정도 개선되었으며, 단어 오인식률은 5.3%정도 증가하였다.

표 2. 히스토그램 pruning에 의한 실험 결과

히스토그램	단어 인식률	문장 인식률	N_{avg}	$x_{RealTime}$
No	93.39	70.26	5,141	2.73
Yes	93.04	69.97	3,268	2.07

표 3은 PLA (phoneme look-ahead) pruning 방법을 적용하였을 때 음소의 평균 지속시간 정도의 값을 사용하는 Δt 의 값을 달리하여 얻은 결과를 나타내고 있다. PLA pruning을 두 프레임마다 적용하였으며, Look-ahead 스큐어를 계산하지 않는 프레임에서는 바로 이전 프레임에서 얻어진 스큐어를 이용하였다. 이 실험에서는 120의 빔 사이즈를 사용하였으며, 히스토그램 pruning은 적용되지 않았다. 실험 결과로부터 큰 값의 Δt 를 사용할수록 pruning의 정도가 심하여 탐색 시간이 감소하였으나 인식 성능은 떨어졌다. $\Delta t = 6$ 의 경우 인식 성능의 별 차이없이 35%의 활성 영역 감소를 얻을 수 있었다.

표 3. Phoneme look-ahead에서 Δt 의 영향

Δt	단어 인식률	문장 인식률	N_{avg}	$x_{RealTime}$
0	93.17	69.89	3,240	1.76
4	93.05	69.35	2,478	1.60
6	92.92	69.31	2,120	1.44
8	92.33	68.24	1,813	1.29

관측 확률의 계산량 감축을 통한 실험 결과는 표 4에 있으며, 150의 빔 사이즈와 히스토그램 pruning, PLA pruning 방법이 적용되었다. VQ를 이용한 방법에서 61개의 코드워드가 사용되었으며, neighborhood를 정의하기 위하여 \ominus 의 값으로 1.5를 사용하였다. 트리 구조의 확률밀도함수에서는 트리가 세개의 레벨로 이루어졌으며(root 레벨 포함), 상위 레벨의 cluster 확률 밀도 함수는 16개의 아들 노드들로 분류되었다. 정확히 계산되어야 할 기우시안 성분들을 찾기 위하여 첫번째 단계에서 5개의 cluster 확률 밀도 함수가 선택되어 갔으며 두번째 단계에서는 30개의 cluster 확률 밀도 함수가 선택되었다. 이 방법들은 인식 성능의 별 차이없이 6% 정도의 탐색 시간을 개선하였다.

표 4. 출력 확률의 계산량 감축 방법의 실험 결과

방법	단어 인식률	문장 인식률	N_{avg}	$x_{RealTime}$
baseline	93.32	70.01	2,425	1.65
VQ	92.53	67.74	2,303	1.56
Tree-PDF	92.73	68.15	2,155	1.55

6. 결론

이 논문에서는 3천단어 규모의 무역상당 관련 음성 데이터베이스를 이용하여 고속 탐색 알고리즘을 개발하였다. 탐색 시간을 개선하기 위하여 크게 두가지 측면, 즉 빔 탐색과 히스토그램, phoneme look-ahead와 같은 pruning 방법들을 이용한 활성 탐색 영역의 감소 방안들, 그리고 연속혼합밀도 HMM에서 전체 탐색 비용의 50-80%를 차지하는 출력 확률의 계산 비용을 줄이기 위한 방법들에 관하여 연구 하였다. 이러한 탐색 알고리즘의 개선을 통하여 고속 워크스테이션을 사용하여 실시간의 약 1.5배의 인식 속도를 얻을 수 있었다.

참고 문헌

- [1] C. H. Lee, E. Giachin, L. R. Rabiner, R. Pieraccini, and A. E. Rosenberg, "Improved Acoustic Modeling for Large Vocabulary Continuous Speech Recognition," *Computer Speech and Language*, Vol. 6, No. 2, pp. 103-107, 1989.
- [2] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, N.J., 1993.
- [3] M. K. Ravishankar, *Efficient Algorithms for Speech Recognition*, PhD Thesis, School of Compute Science, Carnegie Mellon University, 1996.
- [4] V. Steinbiss, B.-H. Tran, H. Ney, "Improvements in Beam Search," *Proc. ICSLP-94*, pp. 2143-2146, Sep. 1994.
- [5] H. Ney, D. Mergel, A. Noll, and A. Paeseler, "Data Driven Search Organization for Continuous Speech Recognition," *IEEE Trans. on SP*, Vol. 40, No. 2, pp. 272-281, Feb. 1992.
- [6] H. Ney, R. Haeb-Umbach, B.-H. Tran, M. Oerder, "Improvements in Beam Search for 10000-Word Continuous Speech Recognition," *Proc. of ICASSP '92*, pp. 13-16, March 1992.
- [7] E. Bocchieri, "Vector Quantization for the Efficient Computation of Continuous Density Likelihoods," *Proc. ICASSP '93*, pp. 692-695, 1993.
- [8] T. Watanabe, K. Shinoda, K. Takagi, E. Yamada, "Speech Recognition Using Tree-Structured Probability Density Function," *Proc. ICSLP-94*, pp. 223-226, 1994.
- [9] J. Tou, R. Gonzalez, *Pattern recognition principle*, Addison-Wesley Publishing Company, U.S.A., 1974.
- [10] 최 인정, 권 오욱, 박 종렬, 김 도영, 정 호영, 은 종관, "자동통역용 한국어 음성 데이터베이스," 제11회 음성통신 및 신호처리 워크샵 논문집, pp. 287-290, 1994.