

형상 형성 제어를 이용한 어휘인식 공유 모델의 가우시안 최적화

안 찬 식*, 오 상 업**

Gaussian Optimization of Vocabulary Recognition Clustering Model using Configuration Thread Control

Chan-Shik Ahn*, Sang-Yeob Oh**

요 약

연속 어휘 인식 확률 분포의 공유 방법에서는 사용될 모델 파라미터들의 초기 추정치를 생성하기 위한 각 문맥들에 대한 음소 데이터가 반드시 필요하지만 이들 음소 데이터에 대한 모델을 구성할 수 없는 단점으로 가우시안 모델의 정확성을 확보하지 못한다는 단점이 있다. 이를 개선하기 위하여 확률 분포의 혼합 가우시안 모델을 최적화하고, 음소 단위로 데이터를 탐색을 지원하는 형상 형성 시스템을 제안한다. 본 논문의 형상 형성 시스템은 확장 facet 분류를 이용하여 사용자에게 음소 단위의 형상 형성 정보를 제공하므로 가우시안 모델의 정확성을 제공한다. 본 논문에서 제안한 시스템을 적용한 결과 시스템 성능에서 어휘 종속 인식률은 98.31%, 어휘 독립 인식률은 97.63%의 인식률을 나타내었다.

Abstract

In continuous vocabulary recognition system by probability distribution of clustering method has used model parameters of an advance estimate to generated each contexts for phoneme data surely needed but it has it's bad points of gaussian model the accuracy unsecure of composed model for phoneme data. To improve suggested probability distribution mixed gaussian model to optimized that phoneme data search supported configuration thread system. This paper of configuration thread system has used extension facet classification user phoneme configuration thread information offered gaussian model the accuracy secure. System performance as a result of represent vocabulary dependence recognition rate of 98.31%, vocabulary independence recognition rate of 97.63%.

▶ Keyword : 공유 모델(clustering model), 가우시안 모델(gaussian model), 형상 형성 제어(configuration thread control)

• 제1저자 : 안찬식 교신저자 : 오상업
• 투고일 : 2009. 12. 09, 심사일 : 2010. 01. 15, 게재확정일 : 2010. 02. 22.
*광운대학교 컴퓨터공학과 박사과정 **경원대학교 IT대학 컴퓨터 미디어 교수
※ 본 연구는 2010년도 경원대학교 지원에 의한 결과임.

1. 서론

모델들을 대상으로 하여 파라미터들을 공유할 수 있는 집합들을 결정하는 공유방법은 모델 파라미터의 집합들을 사용함으로써 보다 강인한 모델들을 추정하며 군집화에 사용될 모델 파라미터들의 초기 추정치를 생성하기 위하여 각 문맥들에 대한 데이터가 반드시 필요하다.[1]

가우시안 모델은 특징 벡터 x 를 2차 이상의 d -차원에서 확률변수로 가지는 가우시안 확률밀도함수를 구성하여 사용한다. 구성되어진 가우시안 모델은 하나의 클러스터링으로 모델링하므로 사용될 모델 파라미터들의 초기 추정치를 생성하기 위한 각 문맥들에 대한 음소 데이터가 반드시 필요하지만 이들 음소 데이터에 대한 모델을 구성할 수 없는 단점으로 가우시안 모델의 정확성을 확보하지 못한다는 단점이 있다. 또한 미리 정의되지 않은 음소와 추가되어진 음소로부터 인식률이 저하되는 문제점이 발생한다.

이를 개선하기 위하여 주어진 표본 데이터 집합의 분포밀도를 하나의 확률밀도함수로 모델링하는 방법을 개선한 밀도 추정 방법이며 복수 개의 가우시안 확률밀도함수로 데이터의 분포를 모델링하는 방법인 혼합 가우시안 모델을 사용하였으며 미리 정의 되지 않은 음소와 추가되어진 음소로부터 인식률이 저하되는 문제를 해결하기 위해 음소 단위로 데이터를 탐색하는 형상형성 제어 시스템을 제안한다.

본 논문에서 제안한 라이브러리는 사용자 하여금 관심 있는 음소를 컴포넌트 단위로 탐색할 수 있고, 컴포넌트들의 집합이 크고, 그 컴포넌트들이 여러 응용 분야에 걸쳐 널리 사용되면서 우수한 모듈성을 갖는 환경에서는 필요한 컴포넌트를 형상 형성 제어(configuration thread control)하고 식별하는 것이 중요한 문제로 제기된다[2][3][4][5].

기존의 어휘인식에서는 일반적인 벡터 값을 데이터베이스를 이용하여 구하므로 탐색 중에 형성되는 음소를 처리하지 못하는 문제점을 제공하지만, 본 논문에서는 라이브러리 구성을 확장된 facet 분류를 응용하여 하부 표현 구조에 관계없이 라이브러리로부터 음소를 관리 및 제어할 수 있도록 하였다. 또한 기존에 연구된 어휘 인식 시스템들이 유용하지만 다른 시스템과의 인터페이스를 제공하지 않으며, 일부 단위만 음소를 관리하므로 포괄적이지 못하여 제한적으로 사용되고 있는 문제점을 처리할 수 있도록 설계하였다.

연속 확률분포의 공유로부터 가우시안 모델 최적화를 위한 라이브러리 시스템을 실험한 결과 빠르고 좋은 인식성능을 확인하였으며 어휘 중속 인식률은 98.31%, 어휘 독립 인식률은

97.63%의 인식률을 나타내어 빠르고 좋은 인식성능을 확인하였다. 제 2장에서는 CHMM 연속 확률분포와 가우시안 모델, 그리고 라이브러리에 대해 설명하고, 제 3장에서는 본 논문에서 제안한 시스템에 대하여 설명한다. 제 4장에서는 제안한 시스템의 실험결과에 대하여 설명하고 제 5장에서 결론을 기술한다.

II. 기존 연구

2.1 연속 확률 분포

수치로 대응되는 확률변수의 개별 값들이 가지는 확률 값의 분포를 확률분포라 한다. 확률변수 X 가 x 의 값을 가지는 확률 $P(x)$ 라 표시한다.[6]

확률분포 함수 중에서 가장 많이 사용하는 분포가 가우시안 확률밀도함수이다. 1차원의 특징 벡터일 경우에는 두 개의 파라미터, 즉, 평균 μ 과 표준편차 σ 이 정해지면 가우시안 확률밀도 함수는 다음과 같이 표현된다.

$$g_{\mu, \sigma}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \dots\dots\dots (1)$$

1차원에서 가우시안 확률밀도함수가 2차 지수 함수의 형태이므로 2차 혹은 다차원의 이차식이 된다. n 차원 가우시안 확률밀도함수는 n 차원 공간에서의 한 점인 중심 μ 이 되고, Σ 는 $n \times n$ 의 가역적인 양의 정부호 대칭 행렬이 되며, X 대신 $X^T \Sigma^{-1} X$ 의 형태를 가진다. 그러므로 2차 이상의 d -차원을 가지는 특징 벡터 x 를 확률변수로 한 가우시안 확률밀도함수는 다음과 같이 표현된다.

$$g_{(\mu, \Sigma)}(x) = \frac{1}{\sqrt{2\pi^d} \sqrt{\det(\Sigma)}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \dots\dots\dots (2)$$

μ 은 가우시안의 중심인 ($d \times 1$)의 평균 벡터이고, Σ 는 ($d \times d$)의 분산-공분산 행렬이다. 여기서, μ 와 Σ 는 $\mu = E[X]$ 와 $\Sigma = E[(x-\mu)(x-\mu)^T]$ 를 만족하며 가우시안 분포의 핵심 파라미터이다.[7]

가우시안 혼합 밀도 함수가 어떠한 연속 확률밀도로도 근사화할 수 있다는 장점을 지니고 있어 남녀의 음성, 주파수 차이 등의 음성에서의 다양한 변이성을 모델링하기에 적합하다.[8]

연속 확률분포의 경우 상태 j , 시간 t 에서 입력벡터 O_t 를 관측할 확률은 다음 식(3)과 같다.

$$b_j(O) = \sum_{k=1}^M c_{jk} N(O_t, \mu_{jk}, U_{jk}), \quad 1 \leq j \leq N \dots\dots\dots (3)$$

M 은 혼합(mixture)의 수이고, C_{jk} 는 상태 j 에서 k 번째 mixture에 대한 가중치이며, $N(O_t, \mu_{jk}, U_{jk})$ 는 가우시안 확률 밀도 함수이다. μ_{jk} 와 U_{jk} 는 각각 상태 j 에서 k 번째 mixture의 평균벡터(mean vector)와 공분산 행렬(covariance matrix)이고 N 은 전체 상태(state)수이다.[9]

2.2 가우시안 모델

가우시안 혼합 모델은 주어진 표본 데이터 집합의 분포밀도를 단 하나의 확률밀도함수로 모델링하는 방법을 개선한 밀도 추정 방법으로 복수 개의 가우시안 확률밀도함수로 데이터의 분포를 모델링하는 방법이다. 단일한 가우시안으로는 모델링 할 수 없는 복수개의 중심점을 가지는 1차원 데이터와 2차원 환형 데이터에 대하여 견고하게 모델링된다.[10]

확률밀도함수는 가우시안 분포뿐 아니라 다른 분포가 될 수도 있다. 가우시안 혼합 밀도는 단지 확률밀도함수를 가우시안 분포로 가정하는 경우이다. 결국 최종적인 전체 확률밀도함수는 M 개의 가우시안 확률밀도함수의 선형 결합으로 식(4)와 같이 표현된다.

$$p(x|\theta) = \sum_{i=1}^M p(x|\omega_i, \theta_i) P(\omega_i) \dots\dots\dots (4)$$

여기서 $p(x|\omega_i, \theta_i)$ 는 데이터 x 에 대하여 ω_i 번째 성분 파라미터 θ_i 로 이루어진 확률밀도함수를 의미하며, $P(\omega_i)$ 는 혼합 가중치로 각 확률밀도함수의 상대적인 중요도를 의미한다. 혼합 가중치를 사전확률과 같은 형태로 α_i 라고 하면 식(5)과 같은 제약조건이 따른다.[11]

$$0 \leq \alpha_i \leq 1, \text{ and } \sum_{i=1}^M \alpha_i = 1 \dots\dots\dots (5)$$

확률밀도함수가 가우시안 분포를 따를 경우 θ_i 는 식(6)와 같은 파라미터 집합이 된다.

전체 모델을 이루는 각 가우시안 성분은 완전대각 또는 정방형 공분산 행렬의 형태를 가질 수 있다. 또한 혼합 성분의 개수는 학습 데이터 집합의 크기에 따라 조절 가능하다.

$$\theta_i = (\mu_1, \mu_2, \dots, \mu_M, \sigma_1, \sigma_2, \dots, \sigma_M, \alpha_1, \alpha_2, \dots, \alpha_M) \dots\dots\dots (6)$$

가우시안 혼합 모델로 데이터의 분포를 모델링할 경우에 혼합 성분의 개수가 충분히 주어지고 적절한 파라미터 값들만

주어진다면 이론적으로는 어떠한 연속적인 분포도 완벽하게 추정하여 모델링할 수 있다.[12]

2.3 형성 형상 제어

유사한 어휘의 선택은 분류 문제이며, 유사성의 정도는 어떻게 집합(collection)을 조직하느냐에 따라 정해진다. 그러므로 집합의 조직과 선택은 이 모델에 있어서 중요하다[13][14].

어휘의 관련성을 부여하는 방법에 따라 enumerative와 facet 방법[15], 어휘처리 수준을 포함하는 분류 방법[16], 그리고 구문 및 의미 분석을 포함하는 방법[17]이 있다.

라이브러리의 예로는 Raytheon Company의 라이브러리, AT & T Pacific Bell사의 C언어 라이브러리 등이 있으며, 이외에도 Prieto Diaz의 라이브러리, EIFFEL, LaSSIE, AIRS, ROSE사 등에서 구축한 라이브러리가 있다.

형상형성 제어는 여러 리비전이나 형상의 공존을 허용할 수 있어야 한다. 이런 지원을 위해서는 각 문서에 이름뿐만 아니라 리비전이나 형상을 구별할 수 있는 레이블을 지원해야 하며, 또한, 형상관리는 형상 C 가 문서 D_i 에 의존적인 모든 문서들이 새로운 리비전을 갖게 되었을 때 D_i 의 새로운 리비전을 생성할 수 있어야 한다. 이는 문서에 영향을 주는 모든 전위 문서들이 이 리비전을 공유할 수 있는가에 대한 무결성을 점검함으로써 얻을 수 있다. 형상 형성 제어는 유도된 새로운 리비전들이 같은 전위 리비전으로부터 유도되었는가를 식별할 수 있어야 하며, 주어진 특정한 형상을 구성하기 위해서 사용되는 문서들 각각의 정확한 리비전들을 찾을 수 있어야 한다.

III. 가우시안 모델 최적화를 위한 형상 형성 제어

3.1 시스템 모델

기존의 어휘인식에서는 일반적인 벡터 값을 데이터베이스를 이용하여 구하므로 탐색 중에 형성되는 음소를 처리하지 못하는 문제점과 하나의 클러스터링으로 모델링하므로 가우시안 모델이 정확성을 확보하지 못한다는 단점이 있다.

이러한 단점을 개선하기 위하여 공유모델의 가우시안 최적화를 위한 형상형성지원 라이브러리를 사용하였다.

라이브러리 시스템은 음소를 컴포넌트 단위로 탐색하고, 컴포넌트들의 집합이 크고, 그 컴포넌트들이 여러 응용 분야에 걸쳐 널리 사용되면서 우수한 모듈성을 갖는 환경에서는 필요한 컴포넌트를 형상 형성 제어하고 식별하며 확장된 facet

분류를 응용하여 허부 표현 구조에 관계없이 라이브러리로부 터 음소를 관리 및 제어한다.

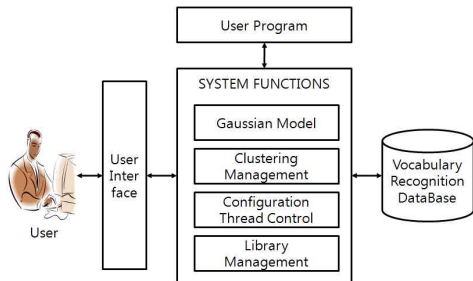


그림 1. 군집화 모델링 시스템 구성도
Figure 1. Cluster modeling system architecture

그림 1에서 가우시안 모델에 의해 생성되어진 음소 모델을 군집하여 재생성된 모델은 검색을 위한 음소단위 확률 모델을 지원하고 형성형상제어를 통해 라이브러리로 관리되어지며 라이브러리 시스템은 사용자로 하여금 관심 있는 음소를 컴포넌트 단위로 탐색 가능하다.

3.2 가우시안 모델

단일 가우시안 출력 확률 밀도 함수를 갖는 3상태 단 음소 모델 초기 집합을 생성하고 훈련한다. 단 음소의 상태 출력 분포는 재 추정하여 훈련된 트라이폰 모델 집합을 초기화하기 위해 복사한다. 동일한 각 음소로부터 유도된 트라이폰들의 각 집합에 대해 대응되는 상태들을 군집화한다. 각 결과 군집에서 대표적인 상태가 선택되고 모든 군집 내의 상태들은 대표 상태로 묶이게 된다. 각 상태의 혼합 요소의 수를 증가시켜 재추정하여 모델의 정밀도를 향상시킨다.[18]

음성신호로부터 특징벡터들을 추출하여 가우시안 값을 추정하고 가우시안 모델을 만드는 과정을 그림 2에 나타내었다.

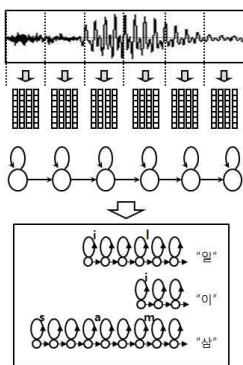


그림 2. 가우시안 모델
Figure 2. Gaussian model

3.3 클러스터링 관리

생성된 가우시안 모델을 군집화하기 위하여 거리 측정법을 사용하였다. 거리 측정에 사용하한 수식은 다음과 같다.

$$d(i, j) = \left[\frac{1}{n} \sum_{k=1}^n \frac{(\mu_{ik} - \mu_{jk})^2}{\sigma_{ik} + \sigma_{jk}} \right]^{\frac{1}{2}} \dots\dots\dots (7)$$

n은 데이터의 차수를 나타내고 μ_{ik} 와 σ_{ik} 상태 s (i 혹은 j)의 가우시안 분포의 k번째 평균과 분산이다. 혼합 거리는 두 가우시안 사이의 겹치는 부분을 측정한다.

$$B_{distance} = \frac{1}{8} (\vec{\mu}_1 - \vec{\mu}_2)^T \times \left(\frac{\Sigma_1 + \Sigma_2}{2} \right)^{-1} \dots\dots\dots (8)$$

$$\times (\vec{\mu}_1 - \vec{\mu}_2) + \frac{1}{2} \ln \frac{\left| \left(\frac{\Sigma_1 + \Sigma_2}{2} \right) \right|}{\sqrt{|\Sigma_1| |\Sigma_2|}}$$

μ_1 과 μ_2 은 각 분포의 평균이며 Σ_1 과 Σ_2 은 공분산이다. 혼합 거리는 비슷한 가중치를 갖는 가우시안들이 조합되기가 더 쉽기 때문에 측정된 형태로 구하게 된다. 가중치 크기 B_{scale} 는 가우시안의 가중치 w_1 과 w_2 의 함수이다.

$$B_{scale} = \sqrt{\frac{w_1^2 + w_2^2}{2w_1w_2}} \dots\dots\dots (9)$$

B_{scale} 는 $w_1 \rightarrow w_2$ 일때, $B_{scale} \rightarrow 1$ 이 된다. 반대로 $w_1 \gg w_2$ 이거나 $w_1 \ll w_2$ 일 때 $B_{scale} \rightarrow \infty$ 가 된다. 이것은 가우시안 모델간의 가중치 차이를 최소화한다.[19]

그림 3은 가우시안 모델들로부터 클러스터링을 통해 공유 모델을 생성하고 관리할 수 있는 형태를 보여준다.

가우시안 모델들을 메모리에 로딩하고 가장 가까운 두 개의 가우시안을 찾는 과정을 반복한 후에 수렴하는지에 따라 계속 할 것인지 종료할 것인지를 결정하게 된다. 가장 가까운 두 개의 가우시안을 찾는 과정은 가우시안 집합에 가우시안이 있는지를 검사하고 남아 있으면 가장 가까운 두 개의 가우시안을 찾는다. 두 개의 가우시안을 찾으면 형성형성 관리를 위한 클러스터링 관리목록에 추가한다.

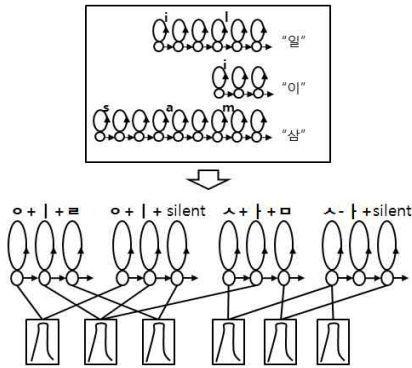


그림 3. 클러스터링 관리
Figure 3. Clustering management

3.4 형상 형성 제어

음소 단위 지원을 위한 형상형성 제어는 음소 탐색 기능을 위하여 컴포넌트 자체 내에 음소에 대한 정보 또는 버전 제어 과정에 도움이 될 수 있는 주석을 포함한다. 즉, 음소의 소재, 음소의 복잡도, 음소 제어에 필요한 절차나 주의사항 그리고 관련 문서 등의 정보를 컴포넌트 내에 작성한다. 음소의 형상 형성 제어에는 이러한 주석 또는 컴포넌트에 기술된 특정 명령어를 가지고 찾도록 한다. 이를 위한 질의어는 사용자가 음소 컴포넌트를 찾는데 필요한 단어를 사용하여 작성하도록 설계하였으며, 이러한 과정은 사용자가 입력한 질의어를 통해 자동적으로 수행되도록 하여 시스템을 관리하고 확장하는 부담을 줄일 수 있게 하였다.

형상형성 제어를 사용하여 각 음소별 표시, 수정, 보고서, 각 음소 관리, 최근 음소 작업 내용 등의 작업을 수행할 수 있으며, 질의 관리는 변동 사항에 대한 음소 관리 기능을 수행하며, 음소 이름, 작업자, 작업일자, 조건 사항 등을 관리한다.

형상형성 제어는 구현한 PhonemeSearch 클래스를 이용하여 다양한 검색 기능을 수행하고, WindowApp클래스와 Dialog 클래스를 사용하여 탐색을 위한 환경을 지원한다.

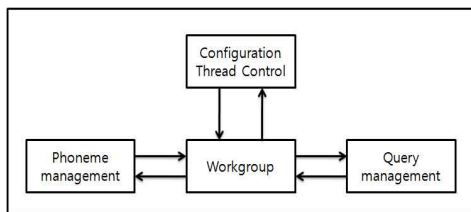


그림 4. 음소 관리, 질의 관리, 그리고 형상형성 제어 시스템과의 연관관계
Figure 4. Phoneme management, Query management, and Configuration Thread Control system Linkage

형상형성 제어 시스템을 위한 PhonemeSearch 클래스는 다음과 같다.

```
class PhonemeRetrieve {
public:
    PhonemeSearch(PTWindowsObject pParent, char drive);
    virtual ~PhonemeRetrieve();
    void Retrieve();
protected:
    virtual int checkPhoneme(struct fsblk& fs) = 0;
    void print(struct fsblk& fs);
    char filePattern[13];
    int startDisk;
    char startPath[MAXPATH];
};
```

형상형성 제어 시스템에서는 검색 메뉴를 위한 기본 메뉴를 설계 및 구축하고, PhonemeSearch 생성자내에서 디렉토리 구조를 트리로 보고 재귀적인 방법으로 각 디렉토리의 컴포넌트를 검색할 수 있도록 하였다. 음소 크기에 의한 검색은 PhonemeRetrieveSize 클래스와 checkFile() 함수를 가지고 파일의 크기를 비교하였다. 다음은 Phoneme RetrieveSize 클래스이다.

```
class PhonemeRetrieveSize : public PhonemeRetrieve {
public:
    enum matchType { lessThan, equalTo, moreThan };
    PhonemeRetrieveSize(PTWindowsObject pParent, char drive, long siz, matchType t);
protected:
    int checkFile(struct fsblk& fs);
    long siz;
    matchType match;
};
```

3.5 라이브러리 관리

본 논문에서는 확장된 facet 방법을 사용하여 라이브러리를 구축한다. 이 방법은 새로운 음소를 추가하기 쉽고 라이브러리 확장에 쉽게 적용된다. 또한, 현재 음소 단위의 계층적 관련성 표현의 자동화는 시스템의 규모가 커지고, 소프트웨어의 규모가 커짐에 따라 복잡한 문제점을 가지고 있다. 이 문제를 해결하기 위해 본 논문에서는 사용자가 사용자 인터페이스를 통해 본 논문에서 제시한 형상형성 제어 시스템 사용할 때에 작업으로 각 컴포넌트들 간의 계층 관계를 관리하여 제한적으로 해결하였다. 컴포넌트가 점진적으로 증가하여 라이브러리의 규모가 증대되면, 이 방법은 문제가 발생할 수 있다. 이 문제 해결을 위해 컴포넌트를 검색하기 위한 find() 함수를 구현하여 컴포넌트를 검색하고, 다른 판단 기준을 만족하는 컴포넌트를 찾는 next() 함수를 호출한다. 이는 라이브러리 관리에서 이용되며, 내부에 Retrieve() 함수를 두어 재귀적으로

원하는 음소를 탐색하도록 하였다.

라이브러리는 지원되는 형상형성제어의 Phoneme Retrieve 클래스에 의해 관리되며, 음소 제어를 위한 컴포넌트를 관리할 수 있으며, 형상형성 제어 지원으로 라이브러리에 음소 컴포넌트를 등록, 검색할 수 있는 메소드와 함수를 지원한다.

IV. 실험결과 및 분석

본 논문에서 제안한 라이브러리 시스템은 다른 시스템과의 인터페이스 관계, 사용자 인터페이스, 탐색 방법, 질의어, 분류구조, 확장성을 고려하여 다른 시스템과 비교 분석하였다.

3장에서 설계·구현한 바와 같이 본 논문에서 제안한 시스템은 phoneme manager와 query manager를 지원하며, 사용자 인터페이스에서 확장 facet 방법을 적용한 형상 형성 정보를 이용한 탐색 방법을 제공하여 음소 관리의 효율을 증진시킬 수 있다. 또한, 본 논문은 확장 Facet 방법을 응용하여 디렉토리 구조를 트리 구조로 인식하여 이를 처리할 수 있는 Retrieve() 함수를 작성하였다. 이것은 기존에 연구된 형상형성 제어 시스템들이 유용하지만 다른 시스템과의 인터페이스를 제공하지 않으며, 일부 단위로만 관리하므로 포괄적이지 못하여 제한적으로 사용되고 있는 문제를 해결한다.

어휘 목록은 서울 시내의 지역명 200개, 지하철역명 100개로 구성하였다. 인식 실험에서는 실험에 참가한 화자가 어휘 목록을 5회 발음하여 총 1500단어를 대상으로 실험을 수행하였다. 어휘는 실내 환경과 잡음 환경에서 이동기기 등에 내장되어 있는 내장형 마이크를 사용하여 16kHz Mono로 녹음하였고, 16bit PCM 양자화를 사용하였다. 실험 어휘는 실내 10명, 실외 5명 등 총 15명의 성인 남성이 참가하였다. 기존의 인식기와 비교 실험을 위하여 캠브리지대학의 HTK(Hidden Markov Toolkit), 코트대학의 Julius, 카네기멜론대학의 Sphinx를 사용하였다.[9]

표 1은 기존의 인식기인 HTK, Julius, Sphinx에서 제한적으로 사용하고 있는 시스템과 비교한 결과를 보였다.

표 1. 평가 결과
Table 1. Test Result

시스템 기준	HTK	Julius	Sphinx	제안방법
사용자 인터페이스	메뉴	메뉴	메뉴	시각적
탐색방법	키워드 매칭	키워드 매칭	키워드 매칭	형상형성 정보이용
질의어	제한된 용어사용	제한된 용어사용	제한된 용어사용	자연어, 시그처제공
분류구조	결정트리	트리 네트워크	렉스트리	확장 Faceted 응용
확장성	DB 확장	DB 확장	DB 확장	확장 Facet

사용자 인터페이스에서 보면 HTK, Julius, Sphinx는 메뉴 방식을 사용하는 반면 제안방법에서는 시각적 효과를 추가하여 그림 5에서 보는 것과 같이 사용자가 쉽게 사용할 수 있도록 하였으며 탐색 방법에서도 형상 형성 정보를 이용하여 기존의 키보드 매칭보다 빠르고 정확한 탐색을 할 수 있었다.

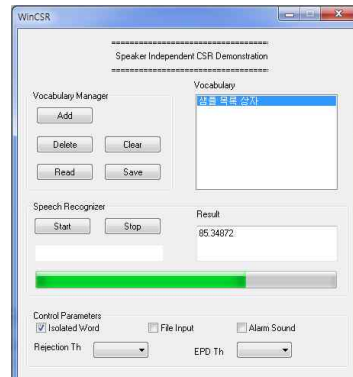


그림 5. 탐색 실행 결과
Figure 5. Search Executed Result

질의어는 자연어처리를 사용하여 제한된 용어를 최대한 줄였으며 분류구조를 트리구조에서 확장 Faceted 응용하여 사용하므로 기존 메모리 사용량을 줄일 수 있었다. 확장성에 있어서는 기존 방법은 DB를 확장하는 방식을 사용하였으나 제안 방법에서는 확장 Facet를 사용하여 DB를 확장해야하는 시간과 노력을 줄일 수 있었다.

표 2 가우시안 수의 따른 인식률
Table 2. Gaussians Recognition Rated

가우시안 수	인식률(%)			
	HTK	Julius	Sphinx	제안방법
1만	98.23	98.75	98.55	98.67
2만	97.51	98.33	98.12	98.21
3만	96.93	97.51	97.08	97.12

표 2은 기존의 방식인 HTK, Julius, Sphinx와 본 논문에서 제안한 방법을 이용한 인식 구조의 가우시안 수의 따른 인식률에 관한 실험 결과이다.

가우시안 수의 따른 인식률 실험에서 HTK, Julius, Sphinx와 제안 방법이 비슷한 성능을 보인 것을 확인할 수 있었다. 가우시안 수가 늘어날수록 인식률이 떨어지는 것을 확인할 수 있었으며 이는 인식할 수 있는 DB가 늘어남에 따라 인식률의 저하를 나타내는 것이다. 제안 방법이 인식률이 가장 높게 나타난 Julius와 비슷한 성능을 보인 것을 확인할 수 있었다.

표 3와 4는 기존의 방식인 HTK, Julius, Sphinx와 본 논문에서 제안한 방법을 이용한 인식 구조의 객체 지향 검색 시스템을 이용한 인식 구조를 실내 환경에서의 실험과 실외 환경에서의 실험을 나타낸다. 실내 환경은 50~55dB에서 실험 하였으며, 실외 환경은 70~75dB의 소음환경 하에서 실험하였다. 결과에서 보는 것과 같이 시스템 성능 평가 결과 어휘 종속 인식률은 98.31%, 어휘 독립 인식률은 97.63%의 인식률을 나타내었다.

표 3. 실내 환경 인식률
Table 3. Indoor Environment Recognition Rate

어휘	인식률(%)			
	HTK	Julius	Sphinx	제안방법
어휘 종속	98.51	97.93	98.15	98.31
어휘 독립	97.77	96.95	97.36	97.63

표 4. 실외 환경 인식률
Table 4. Outdoor Environment Recognition Rate

어휘	인식률(%)			
	HTK	Julius	Sphinx	제안방법
어휘 종속	91.53	90.51	91.74	91.11
어휘 독립	90.07	89.12	90.24	90.01

소음이 50~55dB인 실내 환경 실험에서 어휘 종속 실험과 어휘 독립 실험을 한 결과 어휘 종속 실험에서는 Julius, Sphinx

보다 제안 방법이 높게 나타났으며 HTK 보다는 조금 낮게 나타난 것을 확인할 수 있었으며 어휘 독립 실험에서도 Julius, Sphinx 보다 제안 방법이 높게 나타났으며 HTK 보다는 조금 낮게 나타난 것을 확인할 수 있었다.

소음이 70~75dB인 실외 환경 실험에서는 소음이 적은 실내 환경보다 전체적인 인식률이 저하되는 것을 확인할 수 있었으며 어휘 종속 실험과 어휘 독립 실험을 한 결과 어휘 종속 실험에서는 Julius 보다 제안 방법이 높게 나타났으며 HTK, Sphinx 보다는 조금 낮게 나타난 것을 확인할 수 있었다. 어휘 독립 실험에서도 Julius 보다 제안 방법이 높게 나타났으며 HTK, Sphinx 보다는 조금 낮게 나타난 것을 확인할 수 있었다. 이는 소음이 많은 실내 환경보다 실외 환경에서 성능이 저하되는 것을 확인할 수 있었으며 제안한 방법이 실내 환경에서 좀 더 좋은 성능을 나타내는 것을 확인할 수 있었다.

V. 결론

본 논문은 음소 컴포넌트를 탐색할 수 있는 라이브러리 시스템을 구현하였으며, 라이브러리는 확장된 facet 분류를 응용하여 하부 표현 구조에 관계없이 라이브러리로부터 음소 컴포넌트를 사용자가 탐색할 수 있도록 하여 연속 확률분포의 공유 방법에서 사용되는 각 문맥들에 대한 음소 데이터를 관리하여 가우시안 모델의 정확성을 제공하였다. 또한, 시스템의 확장성을 제공하여 시스템이 추가, 변경되는 경우 부분 수정으로 시스템을 변경할 수 있으며, 다른 시스템과 연계하여 인터페이스를 제공한다.

제안한 방법의 검색 시스템으로 인하여 가우시안 상태의 모델을 최적화하여 혼합 모델을 군집화 함으로써 혼란 중에 나타나지 않는 모델에 대해 인식률을 향상 시킬 수 있는 장점을 확인하였으며 검색 시 속도와 인식률에서 기존 시스템보다 나은 결과를 얻을 수 있었다. 시스템 성능 평가 결과 어휘 종속 인식률은 98.31%, 어휘 독립 인식률은 97.63%의 인식률을 나타냈으며 검색 후 다양한 사용자 지원을 통하여 검색결과를 사용자에게 위한 형태로 변형이 가능하였다.

따라서 공유모델의 가우시안 최적화를 위한 형상형성지원 라이브러리 시스템은 기존 인식 시스템의 높은 인식률을 그대로 적용할 수 있다.

참고문헌

- [1] 안태욱, “혼합 가우시안 군집화를 이용한 상태공유 음향모델 최적화,” 대한전자공학회논문지, 제 42권, SP편 제 6호, 167-176쪽, 2005년 11월.
- [2] 오상엽, 김홍진, 김영선, “UML을 이용한 컴포넌트 버전 제어 시스템 설계,” 한국컴퓨터정보학회논문지, 제 8권 제 1호, 69-75쪽, 2003년 3월.
- [3] 김덕현, 박성주, “확장된 객체지향 데이터 모형을 이용한 소프트웨어 변경 관리 시스템,” 한국정보과학회논문지 제 22권 제 2호, 249-260쪽, 1995년.
- [4] 김행근, “소프트웨어 재사용 지원 정보 저장소 구축,” 한국정보과학회-정보처리학회 공동 특집호, 제 24권 제 11호, 14-31쪽, 2006년 11월.
- [5] 오상엽, 최우승, “버전제어에서 컴포넌트의 형상형성 제어를 위한 객체지향 라이브러리,” 한국컴퓨터정보학회논문지, 제 13권 제 6호, 51-58쪽, 2008년 11월.
- [6] D. Jurafsky and J. H. Martin, “*Speech and Language Processing*,” Prentice-Hall, pp. 21-157, 2000.
- [7] A. S. Manos and V. W. Zue, “*A study on out-of-vocabulary word modeling for a segment-based keyword spotting system*,” Master Thesis, MIT, pp. 19-53, 1996.
- [8] 김동주, 김한우, “문맥기중치가 반영된 문장 유사도 척도,” 대한전자공학회논문지, 제 43권, 제 6호, 496-504쪽, 2006년.
- [9] L. R. Bahl, P. V. deSouza, P. S. Gopalakrishnan, D. Nahamoo, and M. Picheny, “A Fast Match for Continuous Speech Recognition Using Allophonic Models,” InProc. IEEE ICASSP-92, Vol. 1, pp.17-21, 1992.
- [10] L. R. Rabiner, B. H. Juang, “*Fundamentals of speech recognition*,” Prentice Hall, pp. 18-181, 1993.
- [11] T. Jitsuhiro, S. Takatoshi, and K. Aikawa, “Rejection of out-of-vocabulary words using phoneme confidence likelihood,” ICASSP, pp. 217-220, 1998.
- [12] S. Young, D. Kershaw, J. Odell, D. Ollason, Valtcher, P. Woodland, “*The HTK Book*,” Cambridge University Engineering Department, pp. 17-81, 2002.
- [13] Keith E. Golen, “An Object-Oriented Class Library for C++ Program,” *Software -Practice and Experience*, Vol. 17(12), pp. 899-922, 1987.
- [14] 박서영, 김갑수, 명선영, 신영길, 우치수, “객체지향 패러다임에서의 소프트웨어 컴포넌트 분류에 관한 연구,” 한국정보과학회논문지, 제 20권, 제 2호, 879-882쪽, 1993년.
- [15] Ruben Prieto-Diaz and Peter Freeman, “Classifying Software for Reusability,” *IEEE Software*, pp. 6-16, January 1987.
- [16] R. Helm, Y. S. Maarek, “Integrating Information Retrieval and Domain Specific Approaches for Browsing and Retrieval in Object Oriented Class Libraries,” *Proceeding of OOPSLA'91*, pp. 47-61, 1991.
- [17] P. Devanbu, et. al, “LaSSIE : A Knowledge Based Software Information System,” *CACM*, Vol. 34, No. 5, pp. 34-49, 1989.
- [18] K. Demuynck, J. Duchateau, and D. Van Compermolle, “A static lexicon network representation for cross-word context dependent phones,” In Proc. EUROASPEECH, Vol. 1, pp. 143-146, 1997.
- [19] 안찬식, 오상엽, “MLHF 모델을 적용한 어휘 인식 탐색 최적화 시스템,” 한국컴퓨터정보학회논문지, 제 14권, 제 10호, 217-223쪽, 2009년 10월.

저자소개



안 찬 식

2002 : 광운대학교
컴퓨터공학과 공학석사.
2004 : 광운대학교
컴퓨터공학과 박사수료.
관심분야 : 음성인식, 분산처리, 음성/
음향 신호처리



오 상 엽

1999 : 광운대학교
전자계산학과 이학박사.
현 재 : 경원대학교 IT대학
컴퓨터소프트웨어 교수
관심분야 : 소프트웨어공학, 버전관리,
소프트웨어재사용, 형상관
리, 객체지향, 음성인식, 분
산처리, 음성/음향 신호처리