
연속 어휘 인식 시스템에서 어휘 클러스터링 모델의 성능 지원을 위한 검색 시스템

오상엽*

Retrieve System for Performance support of Vocabulary Clustering Model In Continuous Vocabulary Recognition System

Sang Yeob, Oh*

요 약 기존의 연속 어휘 인식 시스템에서는 의사 결정 트리 기반 공유 모델링 방법을 사용하여 인식을 향상시킬 수 있었으나 이들 음소 데이터에 대한 검색을 지원할 수 없는 문제로 인해 시스템 모델의 정확성을 확보하지 못한 다는 단점이 있다. 이를 개선하기 위하여 연속 어휘 클러스터링 모델에서 음소 단위로 확률 모델을 검색할 수 있는 시스템을 모델링하였다. 본 논문에서 제안한 시스템을 적용한 결과 시스템 성능에서 95.88%의 인식을 나타내었다.

주제어 : 어휘 인식, 클러스터링 모델, 음소 모델링, 모델 공유, 검색 시스템

Abstract Established continuous vocabulary recognition system improved recognition rate by using decision tree based tying modeling method. However, since system model cannot support the retrieve of phoneme data, it is hard to secure the accuracy. In order to improve this problem, we remodeled a system that could retrieve probabilistic model from continuous vocabulary clustering model to phoneme unit. Therefore in this paper showed 95.88%of recognition rate in system performance.

Key Words : vocabulary recognition, clustering model, phoneme modeling, tying model, retrieve system

1. 서론

연속 확률 분포 HMM(Hidden Markov Models) 시스템은 어휘 인식을 높이기 위해 일z반적으로 문맥 종속 모델들과 혼합 가우시안 출력 확률 분포를 이용 한다 [13]. 또한, CHMM(Continuous Hidden Markov Models) 기반 연속 어휘 인식 시스템에서는 어휘 데이터의 효율적인 사용을 위해 공유 방법을 적용한다[11].

기존에 사용되는 연속 어휘 클러스터링 모델링은 공유 모델 인식 성능 향상을 위해 Euclidean과 Bhattacharyya 거리 측정을 적용하여 성능 향상을 기존 시스템에 비해 효율적으로 적용하였다[3]. Euclidean 거리 측정은 가중치를 사용하며, 이에 대해 벡터 값을 계

산할 때 확률 값이 저하되는 문제점이 있으며, Bhattacharyya 거리 측정은 오류율을 부가적으로 사용하여 계산이 복잡해지고 속도가 느린 단점이 있다. 이러한 단점을 보완하기 위해 Euclidean과 Bhattacharyya 거리 측정을 혼합하고 가우시안 조합에서 가중치를 사용하여 벡터 값을 계산하여 인식 성능을 향상한 방법이 적용되었다. 그러나 이들 시스템에서는 기본적으로 표준화된 어휘 관리와 이들을 라이브러리에 체계적으로 저장하여 검색을 지원하는 시스템을 제공하지 않으므로 어휘의 접근, 검색, 제어를 효율적으로 수행할 수 없다. 이러한 단점을 개선하기 위하여 기존의 연속 어휘 인식 클러스터링 모델에 검색 시스템을 보완하여 인식 성능을 향상시킬 수 있는 시스템을 제안하였다. 본 논문에서 제안하는

*이 논문은 2012년도 가천대학교 교내연구비 지원에 의한 결과임.(GCU-2012-R180)

*가천대학교 IT 대학 인터랙티브미디어 교수(교신 저자)

논문접수: 2012년 9월 15일, 1차 수정을 거쳐, 심사완료: 2012년 10월 16일

시스템에서 한 어휘의 변화는 다른 어휘에 영향을 미치고, 이들을 사용자가 스스로 처리하려고 할 경우, 많은 개발 시간의 낭비와 오류를 범하게 되어, 어휘 관리에 실패하게 된다. 그러므로 본 논문은 한 시점의 어휘의 변화가 그 이후에 발생한 어휘에 미치는 영향을 분석하여 그 처리 과정을 자동화하였다. 즉, 어떤 어휘의 버전이 새로 생성되었을 때 그 구성요소를 멤버로 포함하는 더 큰 단위의 어휘에게 그 생성 정보를 알려 주며, 한 버전이 내용이 변경될 때 그 버전으로 부터 파생(derivation)된 다른 어휘도 자동으로 파생될 수 있다.

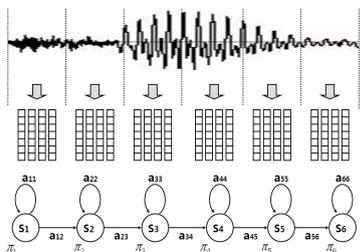
본 논문에서 제안한 혼합 거리측정 방법의 객체 지향 검색 시스템의 성능 검증을 위하여 인식 실험을 수행하였으며, 기존의 방식인 Euclidean, Bhattachayya를 이용한 인식 구조와 본 논문에서 제안한 방법을 이용한 인식 구조 인식 속도를 비교하였다. 본 논문에서 제안한 시스템을 적용한 결과 인식률은 95.88%로 기존의 Euclidean, Bhattachayya 인식률과 비교하여 좋은 인식 성능을 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대해 언급하고 3장에서는 본 논문에서 제안하는 탐색 시스템과 어휘 클러스터링 모델을 설명한다. 4장에서는 제안한 시스템의 인식률 평가를 수행하고 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

2.1 HMM과 CHMM 어휘인식

HMM에서는 정상적인 주파수 범위에서 관측되지 않는 음성의 통계적인 특성을 관측 가능한 벡터 열을 통해 추정하여 음성의 통계적인 변이성을 반영한다[5][8]. 다음 그림 1은 6-states HMM으로 모델링한 상태 천이가 가능한 음성모델을 나타낸다.



[그림 1] 상태의 수가 6인 HMM

그림 1에서와 같이 HMM 모델은 음성 구간의 변이에 의한 상태천이 확률 A_i 와 각 상태에서의 관측 심볼의 출력 확률 B_j , 초기 확률 π_0 로 표현할 수 있으며 다음 식(1)과 같이 정리된다.

$$\lambda = (A, B, \pi) \tag{1}$$

$$A = \{a_{ij}\}, \quad a_{ij} = p[q_{t+1} = j | q_t = i]$$

$$B = \{b_j(k)\}, \quad b_j(k) = p[o_t = v_k | q_t = j]$$

$$\pi = \{\pi_i\}, \quad \pi_i = p[q_1 = i]$$

HMM은 초기 ($t=0$)에 상태 i 의 확률 $\pi_i = \Pr(s_0 = i)$, 상태 i 에서 j 로의 천이 확률 $a_{ij} = \Pr(s_t = i, s_{t+1} = j)$, 상태 j 에서 심볼 k 를 관측할 확률 $b_j(k) = \Pr(x_t = k | s_t = j)$ 로 표현한다. 임의의 음성 특징벡터의 관측열 $O = (o_1, o_2, \dots, o_T)$ 이 사실임을 가정할 때 주어진 N-states HMM 모델에서의 상태열이 $q = (q_1, q_2, \dots, q_T)$ 라면 결국 관측열의 확률은 다음 식(2)와 같이 주어진다 [6].

$$\begin{aligned} P(O|q, \lambda) &= \prod_{t=1}^T P(o_t | q_t, \lambda) \\ &= \sum_{q=Q} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \cdots a_{q_{T-1} q_T} b_{q_T}(o_T) \\ &= \sum_{q=Q} \prod_{t=1}^T a_{q_{t-1} q_t} b_{q_t}(o_t) \end{aligned} \tag{2}$$

초기 상태 $t=1$ 에서 확률 $a_{s_0 s_1}(\pi_{s_1})$ 로 천이가 시작되며, 관측 O_1 는 출력 확률 $b_{s_1}(O_1)$ 로써 생성이 된다. 초기 상태 s_1 에서 상태 s_2 로의 천이는 천이 확률 $a_{s_1 s_2}$ 로 이루어지며, 대응되는 상태 s_2 에서의 관측 O_2 를 생성될 확률은 $b_{s_2}(O_2)$ 가 된다. 이러한 과정은 상태 s_{T-1} 에서 마지막 상태 s_T 로 $a_{s_{T-1} s_T}$ 의 확률로 천이되어 기호 O_T 를 출력 확률 $b_{s_T}(O_T)$ 로 생성할 때까지 계속된다. 이러한 과정을 정의에 의하여 직접 계산하면 모든 시간 $t=1, 2, \dots, T$ 에서는 진행 가능한 상태 수는 N 개가 되어 계산의 복잡도는 $O(N^T)$ 이 된다[9].

이러한 확률계산은 음성 구간에 따라 모델이 지수 함수적으로 증가하는 상태 열을 갖기 때문에 쉽게 계산할 수 없고 계산량이 지나치게 방대해지므로 전향, 후향 알고리즘을 이용하여 HMM 모델의 관측열의 확률을 추정한다[7].

CHMM(Continuous Hidden Markov Model)은 확률 분포 함수 중에서 가우시안 확률 밀도 함수를 가장 많이

사용한다. CHMM에서는 입력된 어휘 신호에서 추출한 특정 벡터를 그대로 사용하며 모델의 파라미터를 추정하기 위해 Maximum Likelihood 방법을 사용한다. 가우시안 혼합 밀도 함수가 연속 확률 밀도로 근사화되는 장점이 있어 남녀의 음성, 주파수 차이 등의 음성에서의 다양한 변이성을 모델링하기에 적합하다.

2.2 Euclidean과 Bhattacharyya 방법

Euclidean 방법은 벡터를 추정하여 거리를 계산하는 측정 방법이며 Bhattacharyya 방법은 오류율을 측정하여 거리를 계산하는 방법이다. 단순 거리 계산을 수행하는 방법이므로 실시간을 요구하는 인식과정에서는 일반적으로 동적 프로그램 기술인 비터비 디코딩 방법을 이용하여 상태경로의 변이와 최적의 모델을 추정하여 인식한다[1].

전향, 후향 확률에 의한 연산을 이용하여 상태경로를 추적하는 경우 인식이 다소 우수한 반면 주어진 모든 상태에서의 출력 기호의 확률을 전부 추정하므로 계산량과 복잡도가 증가하게 된다. 비터비 디코딩의 경우 전향, 후향 확률추정을 이용한 decoding에 비해 인식이 다소 저하되나 연산에서의 부하를 월등히 감소시키므로 일반적인 인식과정에서 이용한다[12].

2.3 검색 시스템

검색 시스템은 데이터베이스에서 소프트웨어의 구성요소를 카탈로깅(cataloging) 하고 검색(retrieving)하기 위해 사용되며, 라이브러리 또는 데이터베이스에 구성요소를 등록시키고 새로운 구성요소를 작성할 수 있어야 한다[2][4].

검색 시스템은 다양한 사용자 사이에서 같은 의미를 가지는 용어(terminology) 차이를 해결하고자 할때 데이터베이스에서 각 텍스트의 유사 단어를 검색하고, 저장된 정보와 사용자의 질의어 사이의 용어 차이 또는 모호성(ambiguity)을 제거하는데 도움을 주기위하여 검색과정 동안 사용된다.

새로운 구성요소가 검색 시스템에 입력될 때, 사용자는 소프트웨어 구성요소와 연관된 정보를 명세하여야 하며, 데이터베이스에 대한 관리도 이루어져야 한다. 소프트웨어 구성요소를 위한 검색 능력은 검색 시스템의 핵심이며, 검색 시스템의 사용은 원하는 정보에 대한 접근을 용이하게 함으로써 정보 수집에 대한 시간과 노력을

단축시키게 된다. 특히 관리할 정보의 양이 기하급수적으로 증가하고 있는 정보화 시대인 오늘날에는 효율적인 정보 검색 시스템에 대한 요구는 증가하고 있다.

정보 검색 시스템은 몇몇의 기본적인 연산을 제공해야 한다. 데이터베이스에 데이터들을 삽입하고, 데이터들을 변경하며, 필요에 따라 삭제할 수 있는 수단을 제공해야 한다. 또한 검색 시스템 지원을 위한 라이브러리는 구성요소의 접근, 탐색, 제어, 보안에 용이한 포괄적인 DBMS이며, 사용자가 구성요소를 생성, 편집, 검증, 합성하기 위한 기능을 제공하며, 확장이 용이해야 한다. 소프트웨어 구성요소의 관련성을 부여하는 방법에 따라 라이브러리는 enumerative와 facet 방법, 어휘처리 수준을 포함하는 분류 방, 그리고 구문 및 의미 분석을 포함하는 방법이 있다.

3. 시스템 모델

클러스터링하여 생성된 모델은 검색을 위한 어휘단위 확률 모델을 지원하고 검색 시스템은 거리측정 방법을 사용하여 모델 유사도를 지원하여 결정 트리에 의해 검색 결과를 나타낸다. 사용자의 요구사항과 일치하는 검색에 실패했을 경우 다양한 검색 방법을 지원하여 검색 효율을 향상시켰으며 확률검색 모델을 이용하였다.

3.1 클러스터링 모델링

클러스터링 모델링에서는 단일 가우시안 출력 확률 밀도 함수를 갖는 3상태 어휘 모델 초기 집합을 생성하고 훈련하며, 어휘의 상태 출력 분포는 Baum-Welch 재 추정을 사용하여 훈련된 트라이폰 모델 집합을 초기화하기 위해 복사하며, 이는 선행 연구를 이용한다[3]. 동일한 각 어휘로부터 유도된 트라이폰들의 각 집합에 대해 대응되는 상태들을 클러스터링하며, 각 결과 클러스터링에서 대표적인 상태가 선택되고 모든 클러스터링 내의 상태들은 대표 상태로 묶이게 된다. 각 상태의 혼합 요소의 수를 증가시켜 재추정하여 모델의 정밀도를 향상시킨다 [10].

시스템의 가우시안 모델은 입력된 어휘를 사용하여 표본 데이터 집합의 분포 밀도를 확률 밀도 함수로 모델링하며, 가우시안 모델이 부족하여 모델의 구성이 원활하지 않을 경우 가우시안 혼합 모델을 사용한다. 가우시

안 혼합 모델은 하나의 확률 밀도 함수로 모델링하는 밀도 추정 방법으로 여러 개의 가우시안 확률 밀도 함수로 데이터의 분포를 모델링하는 방법이다.

확률 밀도 함수는 M 개의 가우시안 확률 밀도 함수를 선형으로 결합한 식(3)과 같이 표현된다.

$$p(x|\theta) = \sum_{i=1}^M p(x|\omega_i, \theta_i) P(\omega_i) \quad (3)$$

식 (3)에서 $p(x|\omega_i, \theta_i)$ 는 입력 데이터 x 에 대하여 어휘에 대한 i 번째 성분 파라미터를 표현하고, θ_i 로 이루어진 확률 밀도 함수를 표현하며, $P(\omega_i)$ 는 혼합 가중치로 표현된다.

3.2 라이브러리 클래스

본 논문에서 제시한 라이브러리 클래스 모델은 어휘를 저장할 수 있는 COMPONENT 클래스와 파생의 자동화에 도움을 주기 위한 DerivationProcedure 클래스 등의 두가지 중요한 클래스를 갖는다. COMPONENT 클래스는 세종류의 부클래스들을 갖는데, 첫번째 부클래스 CompositeCOMP은 다른 소프트웨어 객체들로 구성된 소프트웨어 집합 객체, 즉 부시스템, 시스템, 복합 모듈 등을 인스턴스로 갖으며, 두번째 부클래스인 VERSION은 버전잉에 주제가 되는 소프트웨어 객체의 추상적 정의와 객체의 스냅샷(snapshot)을, 세번째 부클래스인 DerivedCOMP은 다른 소프트웨어 객체로 부터 파생된 객체를 인스턴스로 갖는다. 또한 이들 세가지 부클래스들을 다중 상속받는 CompositeVERSION 클래스, DerivedVERSION 클래스, DerivedCompositeVERSION 클래스 등의 조합 클래스들이 있다. 어휘 등록이 가능한 클래스는 VERSION, CompositeVERSION, DerivationProcedure이며, 본 시스템의 검색 메카니즘은 먼저 어휘의 이름을 받아들여 같은 어휘 이름을 가진 버전들을 찾아낸 후 versionedcomp이라는 변수에 저장된 정보를 기반으로 자신이 원하는 버전의 속성을 작성하여 버전을 검색한다. 검색이 가능한 클래스로는 등록이 가능한 모든 클래스와 DerivedVERSION 클래스, DerivedCompositeVersion 클래스이다. 라이브러리 클래스에서는 DerivedCOMP 클래스와 DerivationProcedure 클래스를 이용하여 하나의 어휘가 다른 어휘로 부터 파생된 것일 때 그 파생의 자동화에 도움을 주며, 어휘의

변화가 다른 어휘에 미치는 영향을 분석하여 처리하도록 지원하였다. 라이브러리 관리를 위해 본 논문에서는 facet 방법을 사용하여 라이브러리를 구축하였으며, 구성요소를 검색하기 위한 search() 함수를 두어 재귀적으로 원하는 구성요소를 검색하도록 하였다.

3.3 검색 시스템

검색 시스템은 시스템 모델의 전반적인 기능 중 가장 중요한 기능으로써, 어휘 관리를 위한 중요한 기능을 제공한다. 검색 시스템은 라이브러리에 구성요소의 추가, 삭제 및 검색할 수 있는 기능을 지원하도록 한다. CompositeVERSION 클래스이나 VERSION 클래스의 객체를 검색하려면 찾고자 하는 어휘의 이름을 이용하여 versionedcomp이 가진 정보가 출력되고, 사용자는 그 정보에 따라 자신이 찾고자 원하는 어휘의 특성을 기록한다. 이때 사용자가 널값을 기록하면, 그 멤버 변수에 대해서는 어떠한 값을 가진 객체도 선택할 수 있으며, DerivedVERSION 클래스와 Derived Composite VERSION 클래스를 이용한다. 시스템 지원을 위한 PhonemeRetrieve 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];
};
```

4. 실험 결과 및 분석

본 논문에서 제안한 혼합 거리측정 방법의 객체 지향 검색 시스템의 성능 검증을 위하여 인식 실험을 수행하였다. 본 연구에서 잡음 처리는 워너 필터를 사용하였으며, 어휘 목록은 서울 시내의 지역명 100개, 지하철역명 50개를 사용하였다. 인식 실험을 위해서는 어휘 목록을 3회 발음하여 총 900단어를 대상으로 실험을 수행하였다. 어휘는 실내 환경과 잡음 환경을 모두 대상으로 하며, 이 동기에 내장되어 있는 내장형 마이크로폰을 사용하여

16kHz Mono로 녹음 하였고, 16bit PCM 양자화를 사용하였다. 실험 어휘는 실내 10명, 실외 5명 등 총 15명의 성인 남성이 참가하였다. 실내 환경은 50~55dB이고, 실외 환경은 70~75dB의 소음환경 하에서 실험하였다. 표 1은 기존의 방식인 Euclidean, Bhattachayya를 이용한 인식 구조와 본 논문에서 제안한 방법을 이용한 인식 구조 인식 속도에 관한 실험 결과이며, 가우시안 수에 따른 인식율을 구하였다. 기존 방법인 Euclidean과 Bhattachayya 방법에 비해 제안한 방법은 인식률이 더 향상되었다.

〈표 1〉 가우시안 수의 따른 인식률

가우시안 수	인식률(%)		
	Euclidean	Bhattachayya	제안 방법
1만	95.14	95.33	96.61
2만	95.37	95.14	96.43
3만	94.27	94.37	95.88

5. 결론

기존의 연속 어휘 클러스터링 모델링에서는 기본적으로 표준화된 어휘 관리와 이들을 라이브러리에 체계적으로 저장하여 검색을 지원하는 시스템을 제공하지 않으므로 어휘의 접근, 검색, 제어를 효율적으로 수행할 수 없는 문제점을 가진다. 본 연구에서는 이러한 단점을 개선하기 위하여 기존의 연속 어휘 인식 클러스터링 모델에 검색 시스템을 보완하여 인식 성능을 향상시킬 수 있는 검색 시스템을 제안하였다. 시스템에서 한 어휘의 변화는 다른 어휘에 영향을 미치고, 이들을 사용자가 스스로 처리하려고 할 경우, 많은 개발 시간의 낭비와 오류를 범하게 되어, 어휘 관리에 실패하게 된다. 본 논문은 한 시점의 어휘의 변화가 그 이후에 발생한 어휘에 미치는 영향을 분석하여 그 처리 과정을 자동화하였다.

제안한 시스템으로 인하여 가우시안 상태의 모델을 최적화하여 혼합 모델을 클러스터링화 함으로써 훈련 중에 나타나지 않는 모델에 대해 인식률을 향상시킬 수 있는 장점을 확인하였으며 검색 시 속도와 인식률에서 기존 시스템보다 나은 결과를 얻을 수 있었다.

참고 문헌

- [1] 김기백, 최종호 (2007), 음성인식 기반 콘텐츠 네비게이션 시스템, 한국컴퓨터정보학회지, 15(1), 99-102.
- [2] 박서영, 김갑수, 명선영, 신영길, 우치수(1993), 객체지향 패러다임에서의 소프트웨어 컴포넌트 분류에 관한 연구, 한국정보과학회 논문지, 20(2), 879-882.
- [3] 안찬식, 오상엽(2010), 공유모델 인식 성능을 위한 효율적인 연속어휘 군집화 모델링, 한국 컴퓨터정보학회 논문지, 15(1), 177-183
- [4] 오상엽, 김홍진, 장덕철 (1996), 버전 제어를 위한 소프트웨어 구성요소의 검색 시스템, 한국정보처리학회 논문지, 3(5).
- [5] 우인성, 신좌철, 강홍순, 김석동 (2007), 다양한 연속 밀도 함수를 갖는 HMM에 대한 우리말 음성인식에 관한 연구, 전기전자학회 논문지, 11(2), 89-94.
- [6] 이호웅, 정희석 (2008), 지능형 홈네트워크 시스템을 위한 가변어휘 연속음성인식시스템에 관한 연구, 한국ITS학회 논문지, 7(2), 37-42.
- [7] 조영수, 이기정, 김광태, 홍재근(1994), "HMM을 이용한 한국어 음소인식", 대한전자공학회 학술발표회 논문집, 16(1), 81-84.
- [8] D. Jurafsky and J. H. Martin (2000), Speech and Language Processing, Prentice-Hall.
- [9] K. Demuynck, J. Duchateau, and D. Van Compernelle (1997), A static lexicon network representation for cross-word context dependent phones, In Proc. EUROSPEECH, Vol.1, 143-146.
- [10] Kris Demuynck, Tom Laureys, Dirk van Compernelle, and Hugo van Hamme (2003), FLaVor:a flexible architecture for LVCSR, In EUROSPEECH, .1973-1976.
- [11] L. R. Rabiner, B. H. Juang (1993), Fundamentals of speech recognition, Prentice Hall.
- [12] S. Ortmanms, A. Eiden, H. Ney, and N. Coenen (1997), Look-ahead Techniques for Fast Beam Search, InProc. IEEE ICASSP-1997, pp. 1783-1786.
- [13] S. Young, D. Kershaw, J. Odell, D. Ollason, Valtcher, P. Woodland (2002), The HTK Book, Cambridge University Engineering Department.

오 상 엽



- 1991년 2월 : 광운대학교 대학원
전자계산학과(이학석사)
- 1999년 2월 : 광운대학교 대학원
전자계산학과(이학박사)
- 2007년 2월 ~ 현재 : 가천대학교
IT대학 인터랙티브미디어학과 교수
- 관심분야 : 버전관리, 형상관리,
음성/음향 신호 처리, 차량 통신

· E-Mail : syoh@gachon.ac.kr