

VCOR를 이용한 효율적인 어휘 최적화 관리

오 상 엽^{*}

요 약

어휘 인식 시스템에서는 처리되는 어휘가 나타나지 않는 미 출현 트라이 폰이 존재 하는 단점이 있으며 이에 따른 신뢰도의 분포를 가지고 있지 않기 때문에 정규화를 수행할 수 없다. 따라서 이를 개선하기 위하여 미등록어 거절 알고리즘에서 사용되는 어휘 관리를 최적화하고 음소 단위로 데이터 탐색을 지원하는 VCOR 시스템을 제안한다. 또한 VCOR에서는 어휘 정보를 효율적으로 제공하기 위해 확장 facet 분류를 이용하여 사용자에게 어휘 단위의 정보를 제공하고, 어휘에 대한 향상된 추적 관리 기능을 제공하여 어휘에 대한 인식의 정확성을 제공한다. 본 논문에서 제안한 시스템을 적용한 결과 시스템 성능에서 어휘 종속 인식률은 97.56%, 어휘 독립 인식률은 96.23%의 인식률을 나타내었다.

Efficient Vocabulary Optimization Management using VCOR

Sang-Yeob Oh^{*}

ABSTRACT

In vocabulary recognition system has it's bad points of processing vocabulary unseen triphone and then no got distribution of confidence measure by cannot normalization. According to this problem to improve suggested VCOR(Version Control for Out-of Rejection) system by out-of vocabulary rejection algorithm use vocabulary management optimization and then phone data search support. In VCOR system to provide vocabulary information efficiently offering for user's vocabulary information using extend facet classification that improved for vocabulary measure management function offering accuracy of recognition for vocabulary. In this paper proposed system performance as a result of represent vocabulary dependence recognition rate of 97.56%, vocabulary independence recognition rate of 96.23%.

Key words: vocabulary recognition(어휘인식), out-of rejection(미등록어 거절), vocabulary database(어휘 데이터베이스), version control(버전 제어)

1. 서 론

인간과 기계와의 인터페이스 수단으로 어휘 인식의 연구가 지속적으로 이루어지고 있다. 그러나 어휘 인식이 우리의 일상생활까지 일반화되지 못하는 이유는 잡음에 의한 낮은 어휘 인식 성능과 미등록어로 인한 오인식 단어에 대한 처리 부족 등을 들 수 있다[1].

어휘 인식 시스템을 구현할 때 구현하고자 하는 인식 시스템이 인식할 대상 어휘를 미리 선정하고 이

에 대한 어휘들을 데이터베이스로 구축한다. 이 데이터베이스를 사용하여 인식할 어휘나 음소 모델을 훈련하게 된다. 이렇게 구현한 인식 시스템은 처음에 정의한 어휘에 적합하도록 어휘나 음소 모델이 훈련되므로 미리 정의된 어휘에 대해서는 인식률이 높게 나타나지만 새로운 어휘를 인식할 필요가 있을 때는 다시 모델을 훈련해야 하는 불편함을 가지고 있다[2].

휴대용 단말기와 스마트 폰의 사용이 급증하면서 어휘 인식의 대한 관심이 부각되고 있지만 미리 정의

* 교신저자(Corresponding Author) : 오상엽, 주소 : 경기도 성남시 수정구 복정동 산65(461-702), 전화 : 031-750-5798, FAX : 02-426-9159, E-mail : syoh@kyungwon.ac.kr

접수일 : 2010년 3월 22일, 수정일 : 2010년 7월 27일
완료일 : 2010년 10월 7일

[†] 중신회원, 경원대학교 IT대학 컴퓨터소프트웨어 교수
* 본 연구는 2010년도 경원대학교 지원에 의한 결과임.

된 어휘에 대한 인식은 어느 정도 인식률을 보이며 사용되고 있다. 하지만 새로운 어휘 어휘의 인식이 어렵게 느껴지고 인식률이 기대만큼 나오지 않아 사용이 많지 않다[3]. 인식 대상이 되는 어휘가 새로 추가되어지면 어휘 인식의 모델 훈련을 새로 해야 하고 기존 어휘와의 분류가 되지 않아 인식률의 저하가 나타난다. 또한 신뢰도 분포가 동일하지 않으므로 균일한 인식 성능을 나타낼 수 없는 단점이 있다. 이러한 단점을 개선하고 어휘 인식을 최적화하기 위하여 본 논문에서는 VCOR(Version Control for Out-of-Rejection) 시스템을 제안한다.

제안된 VCOR 시스템은 첫째, 특징 추출에서 추출된 특징 벡터와 음향 모델을 이용하여 효율적인 데이터 탐색을 위한 모델을 제공한다. 둘째, 미등록어 거절 처리를 위해서는 개선된 리젝션 방법을 이용하여 미리 정의되어 모델링된 어휘에 대해선 인식률을 유지할 수 있지만 정의되지 않은 어휘가 새로 추가되어 모델링된 어휘와 연계되어지는 어휘들에 대해 인식률이 저하되는 것을 향상시켰다. 셋째, 어휘 관리를 최적화하고 어휘 단위로 데이터 탐색을 지원하는 버전 제어 처리에서는 어휘 정보를 효율적으로 제공하기 위해 라이브러리는 확장 facet 분류를 이용하여 사용자에게 어휘 단위의 정보를 제공하고, 어휘에 대한 향상된 추적 관리 기능을 제공하여 어휘에 대한 인식의 정확성을 제공한다.

2. 관련연구

2.1 어휘 인식

어휘 인식에서는 음성-음향학적 지식을 이용하는 방법, 통계적 방법, 인공 지능을 이용한 방법, 신경 회로망을 이용한 방법 등이 널리 연구되었다. 통계적 어휘 인식에서는 아래 그림 1에서와 같이 학습 단계에서 추출된 특징 벡터를 이용하여 기준이 되는 음향 모델 또는 기준 패턴을 구하여 인식을 수행한다.

음향모델은 시간적으로 변화하는 음성신호의 특징을 모델링하기 위해 이용하고 있으며 음향모델링 방법으로 HMM(Hidden Markov Model), 신경회로망(Neural network) 등이 사용된다. 성능의 우수성, 유연성, 확장성 측면에서 HMM 기반의 음향 모델이 사용되고 있다.

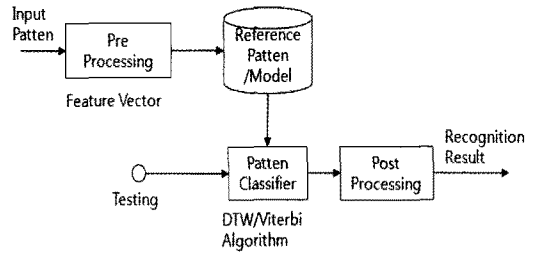


그림 1. 어휘인식 시스템

비터비 알고리즘은 이미 구성되어진 패턴들로부터 부분적인 상태를 이용하여 최상의 단일한 상태를 찾아 탐색하는 알고리즘이며 DTW(Dynamic Time Warping) 알고리즘은 상태열의 길이가 일정하지 않은 두 상태열의 유사도를 측정하는 패턴 정합 알고리즘으로 미리 구성된 패턴 중에서 유사한 패턴을 찾아 인식 결과로 출력한다.

패턴을 인식하기 위한 어휘 인식 구조는 탐색 과정에서 가능한 모든 정보들을 가져오기 때문에 복잡한 언어 모델로 구성되어지므로 음향모델과 언어모델을 사용하는 인식 구조를 갖는다[4].

2.2 미등록어 거절 알고리즘

음소를 기본 단위로 한 어휘 인식 시스템의 성능 향상을 위해서는 각 음소를 다양한 주변 음소 환경 하에서 정확히 모델링하는 것이 중요하며 각 음소의 전후 음소에 따라 각 음소를 달리 모델링하는 3 음소 열(triphone) 모델링 기법을 사용한다.

음소는 전후에 위치하는 음소에 영향을 받기 때문에 세분화하여 사용하는데 이를 3 음소열이라 한다. 3 음소열은 조음화 현상을 모델링하는 인식단위로 그림 2와 같이 모델링한다.

3 음소열은 상태에 대한 확률 값을 이용하며 유사한 음운 환경에 대한 데이터 보완과 신뢰도 향상 때문에 인식 시스템에서 사용된다.

신뢰도는 거절 알고리즘의 척도로 사용되며 어휘 인식기의 출력인 음소 확률과 인식된 음소의 반 음소 확률의 비로 정의 된다. 음소 단위 신뢰도를 구하고 가중 평균을 취하여 단어 단위 신뢰도를 계산하여 사용한다[5]. 음소 단위 신뢰도는 출력 음소 확률과 인식된 음소의 반 음소 확률의 비로 정의 되며 식 (1)에 의해 로그 확률을 표현한다.

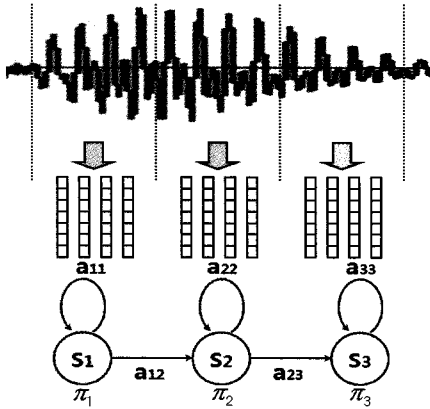


그림 2. 3 음소열 모델링

$$\log pr_a = \frac{1}{M} \sum_{i=1}^{M-1} \log pr_{a_i} \quad (1)$$

$\log pr_a$ 는 반 음소 모델의 평균 로그 확률이고 M 은 반 음소 모델의 수를 나타낸다. 식 (2)는 음소 단위의 신뢰도를 표현한다.

$$cm_p = \frac{\log pr_p - \log pr_a}{|\log pr_p|} \quad (2)$$

cm_p 는 음소 단위의 신뢰도를 계산하기 위한 식으로 $\log pr_p$ 는 단어 구성 음소 모델의 로그 확률이다. $\log pr_a$ 의 반 음소 모델 평균 로그 확률과 $\log pr_p$ 의 단어 구성 음소 모델에 의해서 계산되어진다.

$$CM = \frac{1}{f_{cm}} \log \left(\frac{\sum_{p=0}^{n_p-1} \exp(f_{cm} \cdot cm_p)}{n_p} \right) \quad (3)$$

식 (3)은 가중 평균을 취하여 단어 단위 신뢰도를 나타내며 f_{cm} 은 음의 값을 갖는 가중치이며, n_p 는 단어 구성 음소 수이며 M 은 반 음소 모델의 수를 표현한다[6].

신뢰도는 음소마다 신뢰도의 분포가 동일하지 않기 때문에 단어마다 신뢰도의 분포가 다르고 단어마다 거절 성능이 균일하지 않은 문제점을 갖고 있다.

2.3 버전 제어

시스템이 개발과 유지 보수되는 동안 구성요소인 어휘는 추가, 변경, 삭제되며, 이러한 변경 작업으로 구성요소의 한 버전들을(연속적인) 다음 버전으로 변환시키는 이력 과정(history step)을 제어하여 어휘 인식에서 생산성을 향상시키기 위한 방법으로 버전 제어를 사용한다. 그림 3은 버전 제어 과정을 나타

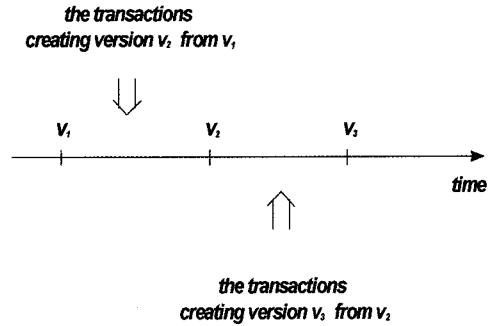


그림 3. 버전 제어 과정

내며, 버전 제어 과정중의 각 이력을 델타 형태로서 관리한다.

어휘 관리를 위한 버전 제어는 프로젝트를 구성하여 프로젝트 구성요소인 어휘 변경 사항에 대한 관리를 지원해야 한다. 버전 제어를 위해서는 기본적으로 표준화된 소프트웨어 구성요소의 작성과 이들을 라이브러리나 데이터베이스에 체계적으로 저장하여 검색이 용이하도록 하여야 하며[7], 이것은 버전 제어를 위한 기반이 된다[8].

2.4 라이브러리

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[9], 어휘처리 수준을 포함하는 분류 방법[10], 그리고 구문 및 의미 분석을 포함하는 방법이 있다.

어휘 처리(lexical processing)를 이용한 시스템은 GURU로서, 어휘 친화력(lexical affinity) 개념과 정보의 양에 기초한 분류 방법을 사용하여 자연어 문서로부터 자동으로 추출된 속성에 따라 부품을 분류하며, 단일 키워드 기반 시스템보다 향상된 정확도를 가진다.

3. 어휘 관리 최적화 시스템 모델

3.1 시스템 모델

미등록어 거절 알고리즘은 음소마다 신뢰도의 분포가 다르기 때문에 어휘마다 신뢰도의 분포가 다르

게 나타나는 단점과 어휘마다 거절 성능이 균일하지 않은 문제점이 있다. 이를 해결하기 위해 그림 4에 VCOR 시스템을 제안한다.

$$D_t = \frac{\sum_{n=1}^N n(C_{t+n} - C_{t-n})}{2 \sum_{n=1}^N n^2} \quad (4)$$

D_t 는 시간 t 에서의 델타 파라미터를 표현하며, C_t 는 시간 t 에서의 멜 캡스트럼의 계수를 나타내고 N 은 델타의 범위 결정 계수를 나타낸다.

미분 값을 사용하는 것은 시간 축 방향의 필터링으로 시간 축 방향으로의 특징 벡터를 얻기 위한 과정이다. 인식을 위하여 주로 사용되는 특징은 MFCC(Mel Frequency Cepstrum Coefficient)와 LPC(Linear Predictive Coefficient)가 주로 사용된다[11].

LPC는 신호의 스펙트럼 및 FFT 캡스트럼으로 얻은 포락을 나타낸다. 선형 분석법 자체가 스펙트럼의 폴(Pole)만을 고려하는 올 폴(all-pole) 모델링에 기반하므로 LPC 캡스트럼으로 얻은 스펙트럼 포락을 사용한다[12].

MFCC는 입력되어진 신호를 안티 앨리어싱 필터(anti-aliasing filter)로 통과시킨 다음, A/D 변환을 거쳐 디지털 신호 $x(n)$ 로 변환한다. 디지털 신호는 고대역 통과 특성을 갖는 디지털 프리엠퍼시스 필터를 통과하게 되어 저주파 신호로 변화하게 된다. 프리엠퍼시스된 신호는 해밍 윈도우(hamming window)를 씌워서 블록 단위의 프레임으로 나누고 프레임 단위로 만든다. 프레임의 크기는 보통 20-30ms를 사용하며 프레임의 이동은 10ms를 사용하여 프레임 단위로 생성한다. 한 프레임의 신호는 FFT(Fast Fourier Transform)를 이용하여 주파수 영역으로 변환하고 주파수 대역을 여러 개의 필터뱅크로 나누어 각각의 बैं크에서의 에너지를 구한다. 밴드 에너지에 로그를 취한 후 DCT(Discrete Cosine Transform)를 취하면 최종적인 MFCC가 얻어진다[13]. 그림 5는 MFCC 추출 과정을 나타낸다.

MFCC 계수는 12개를 사용하며 이와는 별도로 구한 프레임 로그 에너지가 추가적으로 사용되어 인식의 입력으로 사용되는 특징 벡터는 13차 벡터로 구성하여 사용한다. 입력된 신호의 시간 축 방향의 특징 벡터인 12차의 특징 벡터와 위상 값을 지원하는 로그 에너지를 1차로 추가하여 본 연구에서는 13차의 특징 벡터를 사용한다.

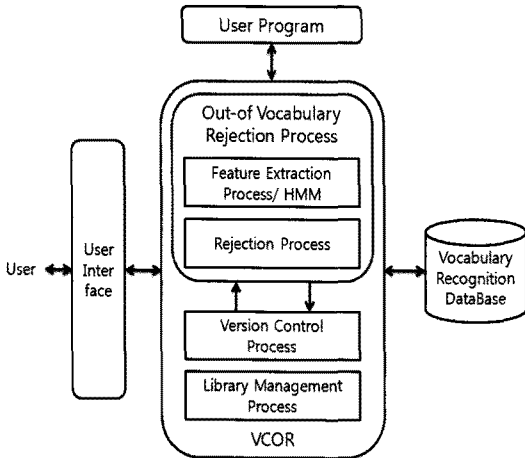


그림 4. VCOR 시스템 구조

VCOR 시스템은 인식에 유용한 성분을 신호로부터 뽑아 특징 추출 처리 과정을 수행하고 각 음소 단위 신뢰도들의 평균과 표준편차를 이용하여 정규화를 통한 미등락어 거절 처리 과정을 수행한다. 어휘 탐색 기능을 위한 컴포넌트 자체 내에 어휘에 대한 정보 지원을 위한 버전 제어를 수행하고 라이브러리 관리 처리에서는 생성되어진 어휘 모델을 군집하여 검색을 위한 어휘 단위 확률 모델을 지원, 관리하며 사용자로 하여금 관심 어휘를 컴포넌트 단위로 탐색하게 한다.

3.2 특징 추출

특성 추출은 인식에 유용한 성분을 신호로부터 뽑아내는 과정으로 정보의 압축, 차원의 감소 과정을 수행하여 특징의 좋고 나쁨을 인식률로 판단한다. 특성 추출 과정에서는 청각 특성을 반영한 주파수 응답 필터 बैं크 분석, 주파수에 따른 대역폭의 증가, 프리엠퍼시스 필터 등을 사용한다. 어휘 신호의 동적 특성을 신호에 반영하기 위하여 캡스트럼 1차, 2차 미분 값을 사용하여 신호를 분석한다. 캡스트럼 1차, 2차 미분 값을 사용한 델타 에너지는 에너지의 시간에 따른 변화율을 나타낸다. 델타 파라미터는 식(4)에 의해 구해진다.



그림 5. MFCC(Mel Frequency Cepstrum Coefficient) 추출 과정

3.3 미등록어 거절

핵심어를 거부하는 오거부는 각 음소 단위 신뢰도들의 통계적 특성의 값이 일정하게 나타나지 않고 불안정하게 나타나므로 오거부가 일어난다. 이를 해결하기 위하여 각 음소 단위 신뢰도들의 평균과 표준편차를 이용하여 정규화시킨 신뢰도를 사용한다.

통계적 분포가 일정하지 않은 분포가 나타나는 현상을 통계적 분포를 정규화하여 사용하며 이전에 계산된 음소 단위 신뢰도의 평균과 표준 편차를 이용하여 각 음소 단위 신뢰도들의 표준 정규 분포를 사용한다[14].

$$mcm_p = \frac{cm_p - TriP_q \cdot mean}{TriP_q \cdot sd} + a \quad (5)$$

식 (5)에서의 $TriP_q, mean$ 는 음소 단위의 신뢰도 평균을 나타내고 $TriP_q, sd$ 는 음소 단위의 신뢰도 표준편차를 나타낸다.

$$NCM = \frac{1}{f_{ncm}} \log \left(\frac{\sum_{p=0}^{n_p-1} \exp(f_{ncm} \cdot mcm_p)}{n_p} \right) \quad (6)$$

식 (6)의 f_{ncm} 은 음소 구성 신뢰도들의 가중치로 사용되며 cm_p 는 기존의 음소 단위 신뢰도를 나타내고 식 (3)에 나타낸 CM 은 정규화하기 전의 음소 단위 신뢰도를 표현한 수식이다. mcm_p 는 정규화된 음소 단위 로그 확률을 표현하며 식 (6)에 표현된 NCM 은 식 (3)을 정규화하여 표현된 수식이다[15]. 미등록어 거절 처리를 위한 알고리즘은 다음과 같다.

```

frame=S_F_SILENCE;
if(m_isActive==0){
  int startX=(m_localFrameX-1)*frameShift;
  if(inStatus==S_EOF && m_sampleEndX - startX
    <outWinSize){
    m_uttEndX=(m_sampleEndX-my_max
      (winSize-frameShift,0))/frameShift;
    m_epdStatus=S_E_POINT;
    return m_epdStatus;
  }
}
  
```

```

else {
  for(i=0;i<outWinSize;i++)out[i]=m_epdSpeech
    [(startX+i)%S_SPEECH_BUF_SIZE];
  frame=Classify(in);
  m_uttEndX=m_localFrameX;
  m_epdStatus=S_SPEECH;
  return m_epdStatus;
}
}
  
```

음소 단위 신뢰도들의 표준 정규 분포를 구하기 위하여 음소 단위 신뢰도의 평균과 표준 편차를 이용하였다. 음소 단위 신뢰도들의 특성 값이 일정하게 나타나지 않기 때문에 통계적 방법을 이용하여 정규 분포를 사용하여 일정한 특성 값을 사용하였다.

3.4 버전 제어

어휘 단위 지원을 위한 버전 제어는 어휘 탐색 기능을 위하여 컴포넌트 자체 내에 어휘에 대한 정보 또는 버전 제어 과정에 도움이 될 수 있는 주석을 포함한다. 즉, 어휘의 소재, 어휘의 복잡도, 어휘 제어에 필요한 절차나 주의사항 그리고 관련 문서 등의 정보를 컴포넌트 내에 작성한다[16]. 어휘의 버전 제어에는 이러한 주석 또는 컴포넌트에 기술된 특정 명령어를 가지고 찾도록 한다. 이를 위한 질의어는 사용자가 어휘 컴포넌트를 찾는데 필요한 단어를 사용하여 작성하도록 설계하였으며, 이러한 과정은 사용자가 입력한 질의어를 통해 자동적으로 수행되도록 하여 시스템을 관리하고 확장하는 부담을 줄일 수 있게 하였다.

버전 제어를 사용하여 각 어휘별 표시, 수정, 보고서, 각 어휘 관리, 최근 어휘 작업 내용 등의 작업을 수행할 수 있으며, 질의어 관리는 변동 사항에 대한 어휘 관리 기능을 수행하며, 어휘 이름, 작업자, 작업일자, 조건 사항 등을 관리한다.

3.5 라이브러리 관리

생성되어진 어휘 모델을 군집하여 검색을 위한 어휘

단위 확률 모델을 지원하고 버전 제어를 통해 라이브러리로 관리되어지며 라이브러리 시스템은 사용자로 하여금 관심 있는 어휘를 컴포넌트 단위로 탐색한다[17].

본 논문에서는 확장된 facet 방법을 사용하여 라이브러리를 구축한다. 이 방법은 새로운 어휘를 추가하기 쉽고 라이브러리 확장에 쉽게 적용된다. 또한, 현재 어휘 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 버전 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. 라이브러리는 지원되는 Phoneme Retrieve 클래스에 의해 관리되어 어휘 제어를 위한 컴포넌트를 관리할 수 있으며, 버전 제어 지원으로 라이브러리에 어휘 컴포넌트를 등록, 검색할 수 있는 메소드와 함수를 지원한다.

4. 실험 결과 및 분석

본 논문에서 제안한 VCOR 시스템을 다른 시스템과의 최적화된 어휘 관리를 고려하여 비교 분석하였다. 3장에서 설계·구현한 바와 같이 본 논문에서 제안한 시스템은 최적화된 어휘 관리를 지원하며 확장 facet 방법을 적용한 버전 정보를 이용하여 미등록된 어휘 관리의 효율을 증진시킬 수 있다. 또한, 본 논문은 확장 facet 방법을 응용하여 디렉토리 구조를 트리 구조로 인식하여 이를 처리할 수 있는 Retrieve() 함수를 사용하였다. 이는 일부 단위로만 관리하여 포괄적이지 못한 어휘에 대해 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 마이크로폰을 사용하여 16kHz Mono로 녹음 하였고, 16bit PCM 양자화를 사용하였다. 실험 어휘는 실내 10명, 실외 5명 등 총 15명의 성인 남성이 참가하였다. 기존의 인식기와 비교 실험을 위하여 캠프리지대학의 HTK(Hidden Markov Toolkit), 교토대학의 Julius, 카네기멜론대학의 Sphinx를 사용하였다[18].

성능 평가를 위해 불검출율을 사용하였으며 불검출율은 미검출율인 MDR(Missed Detect Rate)과 오검출경보확률인 FAR(False Alarm Rate)로 구분된다. MDR은 발생된 핵심어를 제대로 검출하지 못한 경우이고 FAR은 각 핵심어당 오검출 출현횟수를 평가 시간으로 정규화한 것으로 이를 통해 인식률을 확인할 수 있다.

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

탐색 방법에서 버전 제어 정보를 이용하여 기존의 키보드 매칭의 트리구조를 검색하는 시간보다 형상 형성 정보를 이용한 검색이 빠르고 정확한 탐색을 할 수 있었다. 분류 구조를 트리 구조에서 확장 faceted 응용하여 사용하므로 기존 메모리 사용량을 줄일 수 있었다. 어휘 확장성에 있어서 기존 방법은 DB를 확장하는 방식을 사용하여 어휘를 등록할 시 재학습을 하였으나 제안 방법에서는 확장 facet를 사용하여 어휘를 등록할 시 학습을 통하지 않고 DB를 확장하므로 시간과 노력을 줄일 수 있었다.

표 2와 표 3은 기존의 방식인 HTK, Julius, Sphinx와 본 논문에서 제안한 방법을 이용한 실내 환경에서의 실험과 실외 환경에서의 실험을 나타낸다. 실내 환경은 50~55dB에서 실험 하였으며, 실외 환경은 70~75dB의 소음환경 하에서 실험하였다. 결과에서

표 1. 평가 결과

시스템 기준	HTK	Julius	Sphinx	VCOR
탐색방법	키워드 매칭	키워드 매칭	키워드 매칭	형상형성 정보이용
분류구조	결정트리	트리 네트워크	렉스트리	확장 faceted 응용
어휘 확장성	DB확장	DB확장	DB확장	확장 facet

표 2. 실내 환경 인식률

어 휘	인식률(%)			
	HTK	Julius	Sphinx	제안방법
어휘 종속	97.79	96.93	96.15	97.56
어휘 독립	96.77	95.95	95.36	96.23

표 3. 실외 환경 인식률

어 휘	인식률(%)			
	HTK	Julius	Sphinx	제안방법
어휘 종속	90.95	89.89	90.24	90.15
어휘 독립	89.95	88.43	89.55	89.15

보는 것과 같이 시스템 성능 평가 결과 어휘 종속 인식률은 97.56%, 어휘 독립 인식률은 96.23%의 인식률을 나타내었다.

소음이 50~55dB인 실내 환경 실험에서 어휘 종속 실험과 어휘 독립 실험을 한 결과 어휘 종속 실험에서는 제안 방법이 HTK 보다는 조금 낮게 나타났지만 Julius 보다는 0.63% 높게 나타났으며 Sphinx 보다는 1.41% 높게 나타났다. 어휘 독립 실험에서도 제안 방법이 HTK 보다는 조금 낮게 나타났지만 Julius 보다는 0.28% 높게 나타났으며 Sphinx 보다는 0.87% 높게 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 제안 방법이 HTK, Sphinx 보다는 조금 낮게 나타났지만 Julius 보다는 0.26% 높게 나타났으며 어휘 독립 실험에서도 제안 방법이 HTK, Sphinx 보다는 조금 낮게 나타났지만 Julius 보다는 0.72% 높게 나타난 것을 확인할 수 있었다. 이는 소음이 많은 실내 환경보다 실외 환경에서 성능이 저하되는 것을 확인할 수 있었으며 제안한 방법이 실외 환경에서 좀 더 좋은 성능을 나타내는 것을 확인할 수 있었다.

5. 결론

본 논문은 VCOR 시스템을 구현하였으며 특징 추출에서 특징 벡터와 음향 모델을 포함한 HMM을 이용하여 시간적으로 변화하는 음성 신호의 특징을 모델링하므로 효율적인 데이터 탐색을 위한 모델을 제

공하였으며 미등록어 거절 처리에서 새로 추가된 어휘들에 대해 인식률이 저하되는 것을 향상시켰다. 어휘 관리를 최적화하고 어휘 단위로 데이터 탐색을 지원하는 버전 제어 처리에서는 어휘 정보를 효율적으로 제공하기 위해 라이브러리는 확장 facet 분류를 이용하여 사용자에게 어휘 단위의 정보를 제공하고, 어휘에 대한 향상된 추적 관리 기능을 제공하여 어휘에 대한 인식의 정확성을 제공한다.

제안한 방법의 검색 시스템으로 인하여 미등록어에 대한 특징 벡터와 음향 모델을 이용한 효율적인 데이터 탐색을 위한 모델은 확장된 facet 분류를 이용하여 최적화된 어휘를 관리하고 인식률을 향상시킬 수 있는 장점을 확인하였으며 검색 시 속도와 인식률에서 기존 시스템보다 나은 결과를 얻을 수 있었다.

또한, 어휘마다 신뢰도 분포가 동일하지 않게 나타나는 현상을 통계적인 방법인 음소 단위 신뢰도의 평균과 표준 편차를 통해 정규화 분포를 이용하여 어휘마다 신뢰도 분포가 동일하게 분포되어 거절 성능이 균일한 인식 성능을 확인하였으며 어휘 종속 인식률은 97.56%, 어휘 독립 인식률은 96.23%의 인식률을 나타내어 어휘 최적화의 효율적인 관리가 가능하여 인식 성능이 향상된 것을 확인하였다. 따라서 최적화된 어휘 관리를 위한 VCOR 시스템은 기존 인식 시스템의 높은 인식률을 그대로 적용할 수 있다.

참 고 문 헌

[1] 안찬식, 오상엽, "MLHF 모델을 적용한 어휘 인식 탐색 최적화 시스템," 한국컴퓨터정보학회 논문지, 제 14권, 제 10호, 217-223쪽, 2009년 10월.

- [2] Yeonja Lim and Youngjik Lee, "Implementation of the POW(Phonetically Optimized Words) algorithm for speech database," Proc. of ICASSP, pp.89-91, 1995.
- [3] 최승호, "정규화 신뢰도 기반 가변어휘 고립단어 인식기의 저절기능 성능 분석", 한국음향학회지, 제25권, 제2호, pp. 96-99, 2006년.
- [4] D. Jurafsky and J. H. Martin, "Speech and Language Processing," Prentice-Hall, pp. 21-157, 2000.
- [5] 김동주, 김한우, "문맥가중치가 반영된 문장 유사도 척도," 대한전자공학회논문지, 제43권, 제6호, 496-504쪽, 2006년.
- [6] M. G. Rahim, C. H. Lee, B. H. Juang and W. Chou, "Discriminative utterance verification using minimum string verification error (MSVE) training." Proc. of ICASSP 96, pp. 3585-3588, 1996.
- [7] 오상엽, 김홍진, 김영선, "UML을 이용한 컴포넌트 버전 제어 시스템 설계," 한국컴퓨터정보학회논문지, 제8권 제1호, 69-75쪽, 2003년 3월.
- [8] 김덕현, 박성주, "확장된 객체지향 데이터 모형을 이용한 소프트웨어 변경 관리 시스템," 한국정보과학회논문지 제 22권 제 2호, 249- 260쪽, 1995년.
- [9] 오상엽, 최우승, "버전제어에서 컴포넌트의 형상형성 제어를 위한 객체지향 라이브러리," 한국컴퓨터정보학회논문지, 제13권 제6호, 51-58쪽, 2008년 11월.
- [10] 박서영, 김갑수, 명선영, 신영길, 우치수, "객체지향 패러다임에서의 소프트웨어 컴포넌트 분류에 관한 연구," 한국정보과학회논문지, 제20권, 제2호, 879-882쪽, 1993년.
- [11] 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.
- [12] L. R. Rabiner and B. H. Juang, "Fundamentals of speech recognition," Prentice Hall, pp.18-181, 1993.
- [13] K. Demuynck, J. Duchateau, and D. Van Compernelle, "A static lexicon network representation for cross-word context dependent phones," In Proc. EUROSPEECH, Vol. 1, pp.143-146, 1997.
- [14] T. Jitsuhiro, S. Takatoshi, and K. Aikawa, "Rejection of out-of-vocabulary words using phoneme confidence likelihood," ICASSP, pp.217-220, 1998.
- [15] 김철, 이경록, 김진영, 최승호, 최승호, "정규화 신뢰도를 이용한 핵심어 검출 성능향상," 한국음향학회지, 제21권, 제4호, pp. 380-386, 2002년.
- [16] 김행곤, "소프트웨어 재사용 지원 정보 저장소 구축," 한국정보과학회·정보처리학회 공동 특집호, 제24권 제11호, 14-31쪽, 2006년 11월.
- [17] R. Helm and 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.
- [18] S. Young, D. Kershaw, J. Odell, D. Ollason, Valtcher and P. Woodland, "The HTK Book," Cambridge University Engineering Department, pp. 17-81, 2002.



오 상 엽

1999년 광주대학교 컴퓨터과학과 박사
 1993년~현재 경원대학교 IT대학 컴퓨터소프트웨어 교수
 관심분야 : 소프트웨어공학, 버전 관리, 소프트웨어 재사용, 형상관리, 객체지향, 음성 인식, 분산처리, 음성/음향 신호처리