한국어 음성인식 플랫폼(ECHOS)의 개선 및 평가*

권오욱† (충북대), 권석봉(ICU), 윤성락(KAIST), 장규철(KAIST),
김용래(충북대), 김봉완(SiTEC), 김회린(ICU), 유창동(KAIST), 이용주(원광대)

<차 례>

<Abstract>

Improvement and Evaluation of the Korean Large Vocabulary Continuous Speech Recognition Platform (ECHOS)

Oh-Wook Kwon, Sukbong Kwon, Sungrack Yun, Gyucheol Jang,
Yong-Rae Kim, Bong-Wan Kim, Hoirin Kim, Changdong Yoo, Yong-Ju Lee

We report the evaluation results of the Korean speech recognition platform called ECHOS. The platform has an object-oriented and reusable architecture so that researchers can easily evaluate their own algorithms. The platform has all intrinsic modules to build a large vocabulary speech recognizer: Noise reduction, end-point detection, feature extraction, hidden Markov model (HMM)-based acoustic modeling, cross-word modeling, n-gram language modeling, n-best search, word graph generation, and Korean-specific language processing. The platform supports both lexical search trees and finite-state networks. It performs word-dependent n-best search with bigram in the forward search stage, and rescores the lattice with trigram in the backward stage. In an 8000-word continuous speech recognition task, the platform with a lexical tree increases 40% of word errors but decreases 50% of recognition time compared to the HTK platform with flat lexicon. ECHOS reduces 40% of recognition errors through incorporation of cross-word modeling. With the number of Gaussian mixtures increasing to 16, it yields word accuracy comparable to the previous lexical tree-based platform, Julius.

* Keywords: Speech recognition platform, Hidden Markov model (HMM), ECHOS.

* 이 논문은 2005년도 음성정보기술산업지원센터(SiTEC)의 연구비 지원에 의하여 연구되었음.
† 이 논문은 2005년도 충북대학교 학술연구지원사업의 연구비 지원에 의하여 연구되었음.

1. 서 론

새로운 아이디어를 가진 음성인식 연구자의 진입 장벽을 낮추고, 연구역량을 핵심기술에만 집중할 수 있도록 도와주기 위하여 음성인식 플랫폼이 필요하다. 이를 위하여 외국에서는 대학을 중심으로 음성인식 플랫폼을 공개하고 있다. 사실상의 표준 음성 데이터베이스와 공개 소프트웨어를 제공함으로써 새로운 아이디어를 쉽게 검증할 수 있도록 하고 있다.

기존에 공개된 음성인식 플랫폼으로서 HTK(Hidden Markov model Toolkit) [1], Sphinx[2], Mississippi 대학 음성인식기[3], Julius[4], ezCSR[5] 등이 있다. 특히 HTK는 영국 켐브리지 대학에서 개발된 음성인식기의 훈련 및 테스트를 위한 사실상의 표준 플랫폼이다. C언어로 구현되어 있으며, NIST의 벤치마크 테스트에서도 우수한 성능을 보였다. 최근에는 European Telecommunications Standards Institute (ETSI)의 Aurora-2 및 Aurora-3 프로젝트에서의 기준 인식기로 사용되었으며, 2006년부터 대어휘 연속음성인식을 지원하고 있다. 그러나 이들 플랫폼의 내부구조 및 소스코드에 대한 문서가 충분하지 않아서 연구자들이 자신의 알고리듬을 플랫폼 내에 구현하기가 쉽지 않았다.

본 논문에서는 이러한 문제점을 해결하기 위하여 교육 및 연구 목적으로 쉽고 간결한 한국어 음성인식 플랫폼인 ECHOS (Easy Compact Hangeul Object-oriented Speech recognizer)[6][7]를 개발하고 문서화 작업을 추진하였다. ECHOS와 다른 인식 플랫폼과의 비교는 [6]을 참조하기 바란다. ECHOS는 ezCSR의 소스 및 소프트웨어 구조를 바탕으로, 오류 수정, 기능 추가, 매뉴얼 보완, 성능평가 과정을 거쳐서 개발되었다. 쉽고 작으면서 한글 처리가 가능한 객체기반의 구조를 가지며, 표준 템플릿 라이브러리(STL)[8]를 이용한 C++언어로 구현되었다. 이 플랫폼을 이용하여 고립단어 인식, 연속음성인식, 음성 분할 기능을 수행할 수 있다. 사운드카드로부터 직접 입력되는 음성을 인식하는 온라인 모드와 파일에 저장된 음성을 인식하는 오프라인 모드를 지원한다. 이 플랫폼은 응용 프로그램 개발 라이브러리, 음성인식 실험 도구, 음성 파일의 음소단위 분할 도구로도 활용 가능하다.

본 논문은 ECHOS의 초기 버전[7]에서 렉시컬 트리 탐색의 성능을 개선하기 위하여 단일음 단어 처리, 단어그래프 생성 및 최적화, 단어간 모델, 이단계 탐색 방법을 제안하고, 플랫폼의 인식률 및 인식시간을 기존의 공개 플랫폼 중에서 널리 사용되는 HTK 및 Julius와 비교 평가하였다. 플랫 렉시콘을 사용할 경우, ECHOS와 HTK는 비슷한 인식률을 보였으며, 렉시컬 트리를 사용한 경우에 ECHOS는 HTK에 비하여 인식오류는 증가하지만 인식시간은 절반으로 감소하였다. 제시된 개선 방법을 적용한 결과 렉시컬 트리 탐색의 성능을 상대적으로 46% 향상하였으며, 렉시컬 트리를 사용하는 Julius 플랫폼과 비교하여 대등한 수준의 인식률을 달성하였다. 이 결과로부터, 개발된 음성인식 플랫폼은 대어휘 연속음성인식에서 기

존의 플랫폼에 비하여 성능면에서 대등하면서도 쉽고 단순한 구조를 가지고 내부 소스에 대한 문서를 갖추고 있어서, 음성인식 연구자들의 플랫폼으로서 사용되기에 손색이 없을 것으로 생각한다.

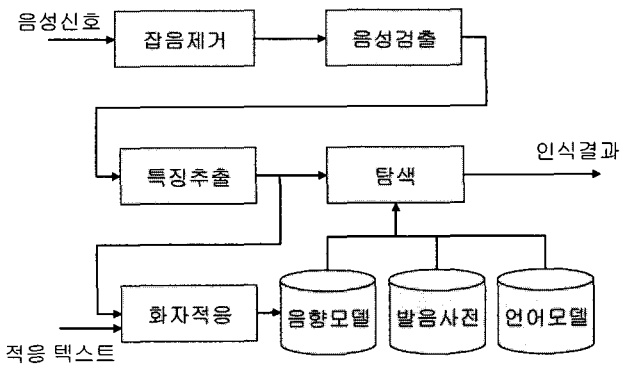
제 2절에서는 ECHOS의 구조 및 기능, 인식엔진 구현방법, 응용 프로그램 인터페이스를 기술하였다. 제 3절에서는 초기 플랫폼의 인식률 향상을 위하여 탐색모듈에 새로이 추가된 기능을 기술하였다. 제 4절에서는 플랫폼의 성능을 평가하기 위하여 인식률 및 인식시간을 HTK 및 Julius와 비교하였다. 제 5절에서 결론을 맺는다.

2. ECHOS

2.1 구조 및 기능

ECHOS는 윈도우 환경에서 마이크 또는 파일을 통하여 입력신호를 받을 수 있으며, 리눅스 환경에서는 파일을 통해서만 입력신호를 받는다. 8/16 kHz 샘플링 주파수와 16 비트 PCM 및 8비트 μ-law 포맷을 지원한다. ECHOS는 1-best, N-best, 단어 그래프(word graph)의 3가지 형태의 인식결과를 제공한다. 인식결과는 단어 아이디(word id), 경계 및 우도(likelihood) 관련 정보를 제공한다.

플랫폼은 <그림 1>과 같이 신호처리, 음성검출, 특징추출, 음향모델, 발음사전, 언어모델, 탐색, 화자적응, 후처리 모듈을 가지며, 각 모듈의 기능은 다음과 같다.



<그림 1> ECHOS의 구조

■ 잡음제거

강인한 음성인식을 위하여 입력신호로부터 배경잡음과 채널잡음을 제거한다

ECHOS는 잡음차감(spectral subtraction), 위너 필터링(Wiener filtering) 방법을 지원한다.

■ 음성검출

입력신호로부터 음성 부분만을 찾아내어 음성인식기로 전달한다. ECHOS는 에너지 기반 음성 검출 알고리듬[9]을 제공한다.

■ 특징추출

입력신호로부터 음성인식에 유용한 특징을 추출한다. ECHOS는 Mel frequency cepstral coefficient (MFCC)[11], perceptually linear prediction (PLP) 계수[12], ETSI 특징[13]을 지원한다.

■ 음향모델

음향모델은 음향특징을 모델링하여 음향단위에 대한 우도를 계산하는 모듈이다. ECHOS는 연속 hidden Markov model(HMM)[9][10]을 채택하며, 대각 행렬 또는 대칭 행렬의 공분산을 지원한다. 음향모델은 HTK 포맷을 따른다. 형성된 음향모델에 대해 훈련된 음향모델이 부족하거나 유사한 확률분포를 가지는 상태의 파라미터를 공유하는 상태공유 기능을 지원한다. 결정트리(decision tree)를 사용하여 음향모델을 갱신하거나 또는 탐색 과정에서 음향모델을 선택할 수 있다. 문맥의존 모델링, 단어내 및 단어간 모델링[15]을 지원한다.

■ 발음사전

인식대상 어휘의 발음을 제공하는 모듈이다. ECHOS는 한글처리를 위하여 발음사전에 한글표제어를 사용할 수 있다. 한글 어휘에 대한 자동적으로 발음기호로 표현해 주는 발음사전 생성기를 제공한다. 한 단어는 여러 개의 발음이 가능하도록 다중발음을 지원한다.

■ 언어모델

언어모델은 연속음성인식을 위하여 문법을 모델링하며, 유한상태망(finite state network; FSN)과 통계적 언어모델[16]로 크게 구분된다. FSN은 인식하고자 하는 단어들의 연결관계를 네트웍으로 표현하는 것으로서 자유도가 낮기 때문에 인식단어 수가 적은 태스크에 주로 사용된다. 통계적인 언어모델은 단어들이 인접하여 발생할 확률을 나타내는 것으로서, 이전 단어의 개수에 따라서 유니그램, 바이그램, 트라이그램으로 나누어진다. ECHOS는 소규모 태스크를 위한 FSN과 대어휘 연속음성인식을 위한 통계적 언어모델인 바이그램, 트라이그램, 클래스 N-gram을 지원한다.

■ 탐색

탐색모듈은 발음사전 및 언어모델로부터 구성된 탐색 네트웍에서 최대 확률을 나타내는 단어열을 찾는 알고리듬이다. 탐색 알고리듬은 크게 소규모 태스크를 위한 FSN 탐색과 대어휘 연속음성인식을 위한 렉시컬 트리 탐색[14][16]으로 나누어진다. FSN 탐색은 인식대상 어휘가 작거나 인식속도가 느리더라도 인식성능에 중점을 두었을 때 사용되는 방식이다. 렉시컬 트리 탐색은 인식대상 어휘가 많거나 인식성능은 떨어지더라도 빠른 인식속도가 요구되는 태스크에 주로 사용된다.

ECHOS는 FSN 탐색과 트리기반 탐색 알고리듬을 모두 지원한다. 인식률 향상을 위하여 이단계 탐색 기법을 지원한다. 먼저 1단계에서 바이그램을 이용하여 탐색 네트웍을 탐색하여 단어그래프를 생성하고, 2단계에서 트라이그램을 이용한 스택 디코딩을 통하여 보다 정확한 인식결과를 얻는다. 전향 탐색과정으로부터 단어 단위의 경계(segmentation)정보와 우도에 대한 정보를 얻는다. 음소 및 상태 단위의 분할 정보와 우도 정보는 1-best 인식결과와 음성신호를 비터비 정렬(Viterbi alignment)함으로써 얻어진다.

음소 및 상태 단위의 분할 정보와 우도 정보를 얻기 위하여 1-best 인식결과와 음성신호를 비터비 정렬하는 기능을 갖는다. 단어그래프로부터 사후확률에 의한 단어의 신뢰도(confidence measure)[17]를 계산한다. 이들 정보는 발화 검증 등에 의한 성능 개선에 활용될 수 있다.

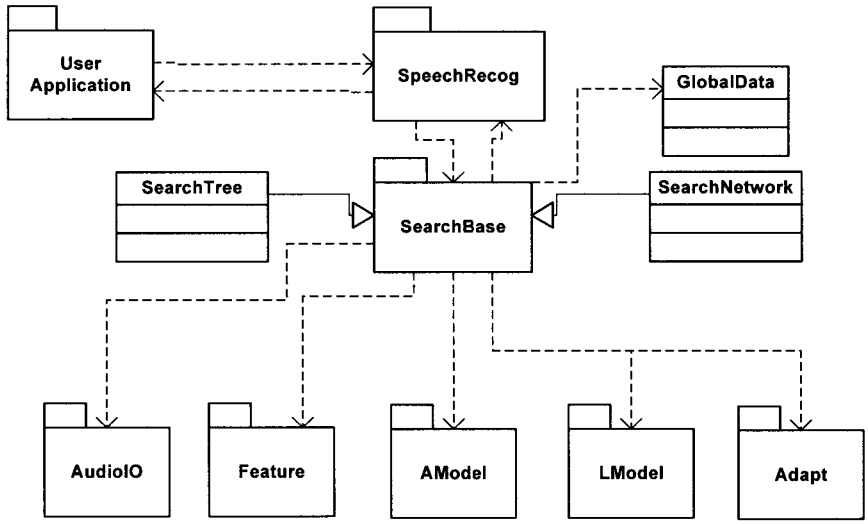
■ 화자적응

화자의 변화에 따라서 음향모델을 변경하여 인식률을 개선하는 모듈이다. ECHOS는 지도적응(supervised adaptation)만을 지원하며, Maximum a posteriori (MAP) 추정 [16] 및 maximum likelihood linear regression (MLLR) [16]에 의한 화자적응 알고리듬을 구현하고 있다.

2.2 소프트웨어 구현

ECHOS는 <그림 2>와 같은 클래스로 구성된다. SpeechRecog 클래스는 사용자 프로그램과의 인터페이스를 담당한다. SearchBase 클래스는 탐색모듈. 인식에 필요한 모든 모듈을 관리하고, 탐색 알고리듬에 따라서 해당하는 탐색객체를 호출한다. SearchTree 클래스는 대어휘를 위한 렉시컬 트리를 구성하여 탐색한다. SearchNetwork 클래스는 소규모 또는 중규모 어휘를 갖는 음성인식을 위하여FSN을 구성하여 탐색한다. AudioIO 클래스는 사운드카드 또는 파일로부터 음성을 읽어 들인다. 대략적인 끝점검출 기능도 동시에 수행된다. Feature 클래스는 입력신호로부터 잡음을 제거하고 특징을 추출한다. 정교한 끝점검출이 사용되어 탐색모듈에게 음성입력 완료를 알려준다. AModel 클래스는 음향모델을 읽어들이고, 입력

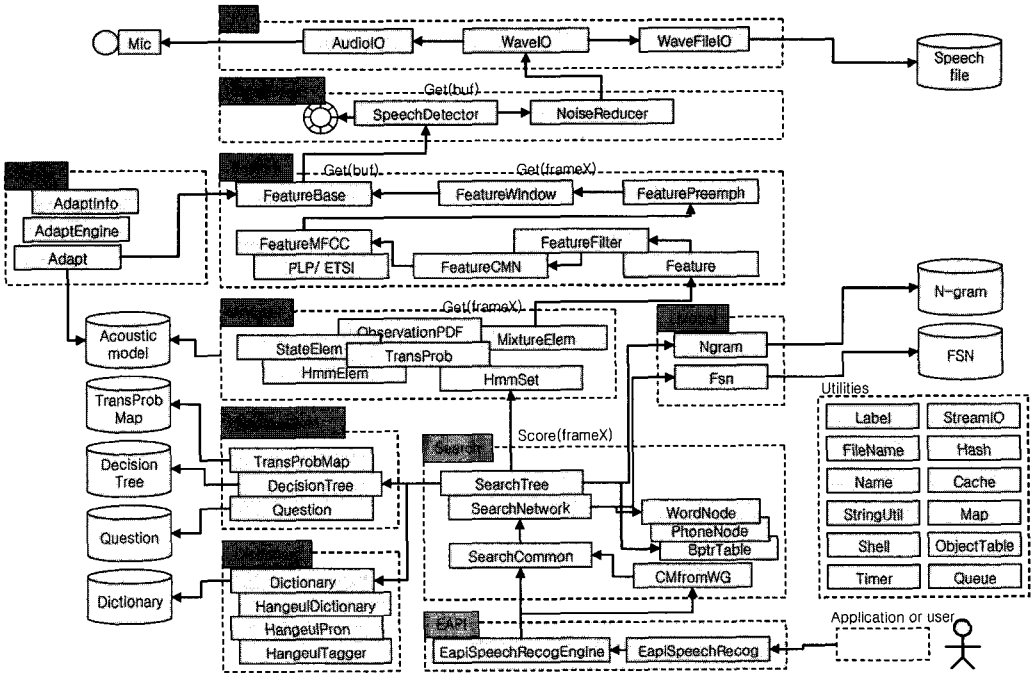
된 특징에 대한 로그확률을 계산한다. LModel 클래스는 언어모델을 읽어들이고, 이전의 단어열이 주어질 때 현재단어의 로그확률을 계산한다. Adapt 클래스는 화자적응을 위하여 관련 통계값을 누적하고 이를 이용하여 음향모델을 변경한다. 주요 모듈 및 보조 모듈을 포함한 ECHOS 전체의 상세한 클래스 다이어그램은 <그림 3>과 같다.



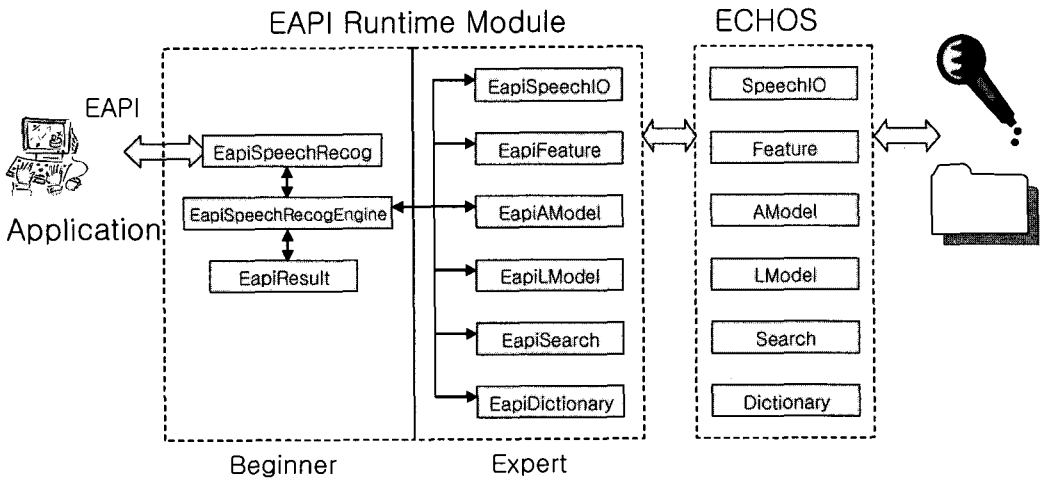
<그림 2> 클래스 다이어그램

2.3 EAPI

ECHOS 응용 프로그램 인터페이스(EAPI)는 ECHOS와 응용프로그램의 인터페이스를 나타낸다. EAPI 규격은 사용자 수준에 따라서 두 단계로 제공된다. EAPI는 인식 플랫폼을 제어하거나, 모듈이 개별적으로 사용될 수 있는 인터페이스를 제공한다. <그림 4>는 EAPI의 구조를 나타낸다. EAPI Runtime module은 인식엔진을 제어하거나 관련 정보를 접근하고 개별 모듈에 대한 인터페이스를 제공하기 위하여 인식엔진에 덧붙여진 것이다. EAPI의 사용 방법을 보이기 위하여 음성입출력, 특징추출, 언어모델, 잡음제거, 음성검출, 간단한 음성인식 방법의 예제를 포함하고 있다.



<그림 3> 세부 클래스 다이어그램



<그림 4> EAPI의 구조

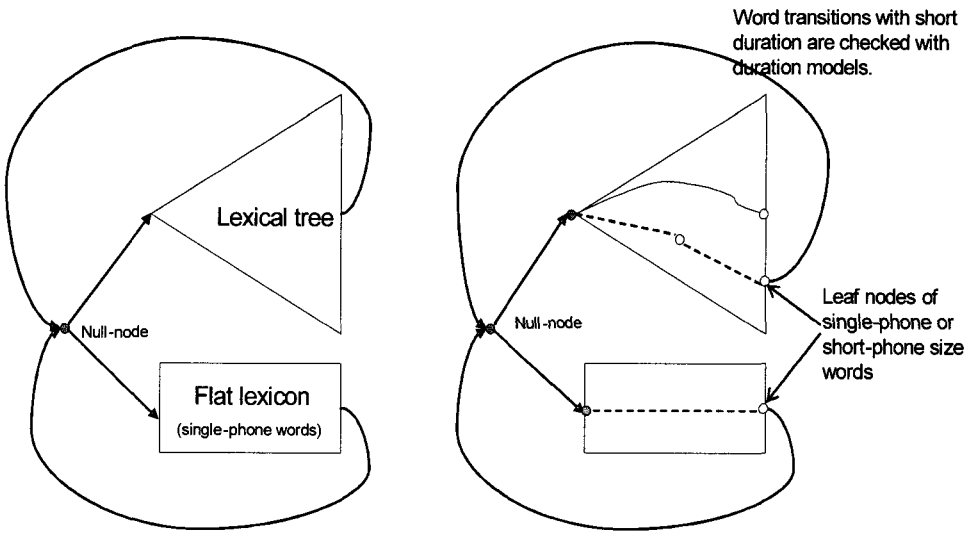
3. 렉시컬 트리 탐색 개선

3.1 짧은 단어 처리

ECHOS는 대어휘 연속음성인식에 사용될 수 있는 네트웍으로 플랫 렉시콘[16]과 렉시컬 트리[16]를 지원한다. 플랫 렉시콘은 모든 단어를 병렬로 네트웍을 구성하기 때문에 많은 음소노드들이 필요로 하고 단어 천이 부분에서도 많은 연산을 해야 하기 때문에 빠른 인식속도가 요구되는 연속음성인식기에는 적합하지 않다. 따라서 대어휘 연속음성인식기에서는 주로 렉시컬 트리를 사용하며, ECHOS에서는 대어휘 연속음성인식에서의 메모리를 줄이기 위하여 단어사전으로부터 구성된 하나의 렉시컬 트리만을 이용하여 탐색한다.

렉시컬 트리는 메모리 사용은 효율적이지만, 언어모델 확률값의 적용 지연과 단어간 모델링 구현의 복잡성 등의 문제가 있다. 이러한 문제점을 해결하기 위하여, 트리 복사 알고리듬[16] 등이 제안되었다. 계산량 및 메모리 증가를 억제하면서도 인식률을 개선하는 다른 방법으로서, 렉시컬 트리의 잎사귀노드(leaf node)로 진입할 때 언어모델을 적용하고, 단일음 단어들을 플랫 렉시콘의 형태로 특별히 처리하는 방법이 제안되어 있다[16]. 이것은 언어모델을 가능하면 조기에 적용함으로써 음향모델 스코어가 낮은 정확한 가설(correct hypothesis)이 가지치기되는 것을 막을 수 있기 때문이다.

단일 렉시컬 트리를 이용하여 탐색하는 경우에는 head노드에서 누적 확률 중첩 문제가 발생하여 인식성능이 떨어진다. 언어모델을 적용할 때, 렉시컬 트리에서는 head 노드로 천이된 단어가 무엇인지 결정할 수 없기 때문에 단어의 천이가 일어나는 부분에서 언어모델을 적용할 수 없다. 특히 짧은 음소열을 갖는 단어의 음소노드에서는 심각한 누적 확률 중첩문제를 야기한다. 여기서 누적 확률 중첩 문제란 입력으로 들어온 특징벡터가 짧은 단어에 대해 그 단어의 발성이 아님에도 불구하고 높은 확률을 갖게 되어 단어의 천이가 빨리 일어나게 되는 현상이다. 음소열이 짧기 때문에 금방 그 단어에 대해 높은 확률을 가지고 다음 단어로 천이가 이루어지고 다른 경로로 들어온 인식경로를 끊어 버리는 현상을 초래하여 인식성능의 저하시킨다. 이를 막기 위하여 ECHOS에서는 <그림 5>(a)와 같이 단일 음소로 이루어진 단어에 대해서는 렉시컬 트리를 구성할 때 별도의 병렬적인 구조를 갖도록 설계하여 누적 확률 중첩문제를 해결하였다. 또한 <그림 5>(b)와 같이 지속시간 모델을 두어 단일음 단어 또는 음소 개수가 작은 단어의 단어간 천이를 2~3 프레임 정도 지연하여 적용하였다.



<그림 5> 단일 렉시컬 트리에 의한 문제를 해결하기 위한 방법. (a) 단일음 단어 트리 추가 (b) 지속시간 모델에 의한 짧은 단어의 천이 보류.

ECHOS는 두 가지치기 방법을 지원한다. 첫째는 이전의 플랫폼에서와 같이 빔 폭을 사용하는 것이다. 둘째로는 백포인터 테이블에 저장되는 음소노드의 개수를 제한하는 것이다. 이 두 가지를 적절히 결합하면 인식시간을 절약할 수 있다. 첫 번째 방법이 디폴트로 사용된다.

3.2 단어간 모델링

단어간 트라이폰 모델링[15]은 단어간의 경계가 불분명하고 강한 조음효과가 발생하는 연속음성인식의 성능 향상에 크게 기여한다. 사람이 연속된 두 단어를 연속해서 발성을 할 때 단어 사이에 묵음 구간을 두고 발성할 수도 있지만 묵음 구간이 없이 연속적으로 발성할 수 있다. 따라서 연속된 두 단어 사이에 묵음 모델을 적용할 뿐만 아니라 단어간 모델(cross-word model)를 적용한다. 또한 단어간 모델은 음향모델은 크게 증가하지만 계산량은 증가분이 음향모델 증가비율에 비하여 상대적으로 작다. 잘 훈련된 음향모델을 사용하면 가지치기에서 더 큰 빔폭을 사용할 수 있어서 탐색공간을 줄이기 때문이다.

단어간 모델을 적용하기 위하여 모든 단어간 모델을 확장하는 경우 음소노드의 개수가 크게 증가한다. 따라서 잎사귀 노드(leaf node)에서는 우측 문맥 음소가 존재하는 경우에만 단어간 트라이폰 모델을 사용하였고, 헤드 노드(head node)에서는 최적의 좌측 문맥 음소에 따라서 단어간 트라이폰 모델을 동적으로 할당하였다.

3.3 이단계 탐색

ECHOS는 이단계 탐색[16]을 지원한다. 이단계 탐색은 시간동기 비터비 알고리듬을 이용한 1단계 전향 탐색에서 간단한 음향모델 및 언어모델을 이용하여 N-best 또는 단어 그래프[18][19]의 형태로 인식결과를 일차로 구하고, 2단계 후향 탐색에서 더 정확하고 복잡한 모델을 이용하여 재인식한다. ECHOS는 전향 탐색에서 바이그램 언어모델을 사용하여 단어 그래프를 얻고, 후향 탐색에서 A* 알고리듬과 트라이그램 언어모델을 사용하여 1-best 또는 N-best 인식문장을 얻었다. 두 단계 모두 음향모델은 동일한 것을 사용하였다. 후향 탐색은 단어 그래프 또는 백포인터 테이블과 같은 작은 탐색 공간을 사용하기 때문에 전향 탐색 시간에 비하여 무시할 정도로 빠르다. 전향에서 바이그램을 가지고 탐색을 하여도 백포인터(backpointer) 테이블과 단어 그래프에는 실제 인식해야 될 정답이 포함되어 있을 가능성이 높기 때문에 전향 탐색에서는 덜 정교한 모델을 가지고 인식을 하고 후향 탐색을 할 때는 탐색공간이 줄기 때문에 보다 정교한 모델을 가지고 인식하는 것이 인식성능과 속도를 동시에 만족시킬 수 있다.

단어 단위로 분할된 구간에서 단어 식별 결과의 후보가 여러 개 생기는 경우에, 이 후보들을 이들의 확실성의 척도와 함께 배열하여 나타낸 것을 단어 그래프(word graph)라고 한다. 이는 다단계 탐색, 발화 검증 또는 새로운 언어모델 적용에 활용될 수 있다. 단어 그래프는 백포인터 테이블로터 직접 구해지기 때문에, 후보 단어로 구성된 많은 에지(edge)를 갖는다. 따라서 단어 그래프를 간단한 것으로 감축하는 작업이 필요하다. 단어 그래프는 가능한 단어열 또는 에지의 개수를 줄이거나, 단어그래프의 오류율이 최소가 되도록 최적화된다. ECHOS에 구현된 단어 그래프 최적화 방법[21]은 펼치기(unfolding), 단어경계 최적화, 가지치기, 노드 병합, 서브그래프 병합 알고리듬이다.

4. 성능 평가

4.1 단어간 모델

대어휘 연속음성인식에서의 성능을 평가하기 위하여 SITEC에서 제작한 Dict01 음성 데이터베이스[20]를 사용하였다. Dict01은 사무실에서 PC를 사용하는 환경에서 16kHz/16비트로 녹음되었다. 음향모델 학습에 400명을 사용하고 인식에 10명의 화자로 이루어진 1,050 문장을 사용하였다. 발음사전의 단어수는 8,670개이고, 언어모델은 Dict01 말뭉치로부터 학습된 바이그램 및 트라이그램이다. 특징벡터는 12차 MFCC와 에너지 및 그에 해당하는 델타, 델타-델타를 포함한 39차 벡터를 사용

하고, 켑스트럼 평균 차감(cepstral mean subtraction)을 적용하였다. 음향모델은 1개의 가우시안을 갖는 트라이폰 모델을 사용하였다.

먼저 ECHOS의 인식과정이 정상적인 동작을 하고 있는 것을 확인하기 위해 HTK [21]와 동일한 네트웍 구조와 파라미터를 사용하여 인식성능 테스트를 하였다. HTK에서는 렉시컬 트리를 지원하지 않기 때문에[1] 먼저 ECHOS에서 플랫 렉시콘 네트웍에서 단어간 모델을 사용하지 않고 파라미터를 동일하게 설정하였다. 그리고 나서 ECHOS에서 지원되는 렉시컬 트리에서 인식성능 테스트를 하였다. <표 1>은 ECHOS와 플랫 렉시콘을 사용한 HTK의 인식성능을 비교한 것이다.

동일한 네트웍 구조를 가지는 플랫 렉시콘에서는 거의 같은 인식률(1행과 3행)을 보이고 있지만 인식시간은 4배 이상 증가함을 보이고 있다. 인식시간의 차이는 ECHOS의 객체지향적 구조와 C++/STL을 사용한 코딩, 음소노드 단위의 가지치기, 단어간 천이시에 바이그램 확률이 적용으로부터 기인한다. HTK에서는 탐색 네트워크의 상태간 천이시 이미 언어모델 값이 반영되어 있다.

ECHOS의 렉시컬 트리 탐색은 플랫 렉시콘을 사용한 HTK보다 인식시간은 50%로 감소하였고, 40%정도의 상대적 에러율 증가를 나타내고 있다. 이는 ECHOS가 단일 렉시컬 트리를 사용하고, 단어간의 천이가 일어나는 순간에 언어모델이 적용되는 플랫 렉시콘 탐색과는 달리 렉시컬 트리 탐색에서는 인식될 단어가 결정되는 단어 끝부분에서 언어모델이 적용되어, 단어가 시작되는 노드에 잘못된 누적확률이 적용되거나 인식되어야 될 단어가 먼저 가지치기(pruning)되는 경우가 발생하기 때문이다.

<표 1> 플랫 렉시콘을 사용한 HTK와 ECHOS의 성능 비교

플랫폼	단어간 모델	탐색 네트웍	인식률 (%)	인식 시간 (초/문장)
HTK	no	Flat-lexicon	84.9	7.1
HTK	yes	Flat-lexicon	92.1	12.0
ECHOS	no	Flat-lexicon	85.2	33.5
ECHOS	no	Lexical tree	75.1	3.5
ECHOS	yes	Lexical tree	85.5	15.8

<표 2>는 HTK와 ECHOS에서 같은 입력음성에 대한 플랫 렉시콘에서의 인식결과를 나타내고 있다. 전체적으로 인식결과에서 조금 차이를 보이고 있지만 인식성능은 비슷하다. <표 2>를 보면 인식 단어열도 조금 다른 문장이 존재하고, 인식된 단어 구간도 조금씩 차이를 보이고 있다. ECHOS는 음소노드를 기본 단위로 해서 인식을 하고 HTK는 state를 기본 단위로 해서 인식하기 때문인 것으로 분석

1) HTK의 현재 버전은 3.3이며, 3.4버전은 대어휘 연속음성인식을 위한 렉시컬 트리를 지원하는 것으로 알려져 있으며, 현재 알파 테스트 중임.

된다. 단어 인식률이 같은 구간임에도 불구하고 조금 차이를 보이는 것은 ECHOS는 언어모델 확률을 이전 단어에 포함하여 저장하기 때문이다.

<표 2> HTK와 ECHOS의 인식결과 비교

HTK 인식결과	ECHOS 인식결과
"./testData/set010001.rec"	"SpeechDB/Dict01/TestMfc/set010001.mfc"
24500000 28100000 그는 -2288.247559	24500000 28100000 그는 -2293.540039
28100000 32800000 동안 -2676.362793	28100000 32800000 동안 -2683.195313
32800000 37800000 선생님 -2991.137207	32800000 37800000 선생님 -2995.767578
37800000 42200000 말씀 -2624.465820	37800000 42200000 말씀 -2630.191406
42200000 44300000 잘 -1251.460571	42200000 44300000 잘 -1257.294922
44300000 47700000 듣고 -1950.922729	44300000 47800000 듣고 -2019.054688
47700000 50200000 오후 -1574.850952	47800000 50200000 오후 -1519.210938
.	.
"./testData/set010002.rec"	"SpeechDB/Dict01/TestMfc/set010002.mfc"
25100000 26700000 그 -985.695374	25100000 27100000 그 -1225.338867
26700000 27300000 후 -407.829712	27100000 27300000 후 -179.035156
27300000 31000000 조용히 -2314.850342	27300000 31100000 조용히 -2374.277344
31000000 34600000 내용을 -2140.939209	31100000 34600000 내용을 -2092.422852
34600000 39800000 보면 -2969.478516	34600000 39800000 보면 -2975.235352
39800000 44400000 다음과 -2734.761230	39800000 44500000 다음과 -2801.250000
44400000 48000000 같다 -2084.654297	44500000 48700000 같다 -2470.886719
48000000 49800000 어 -1172.036743	.
.	
"./testData/set010003.rec"	"SpeechDB/Dict01/TestMfc/set010003.mfc"
26200000 28200000 그 -1207.030518	26200000 28200000 그 -1212.291016
28200000 34400000 사이에 -3392.146973	28200000 34400000 사이에 -3397.743164
34400000 38100000 문을 -2281.131348	34400000 38100000 문을 -2285.939453
38100000 40600000 열고 -1434.741577	38100000 40600000 열고 -1440.447266
40600000 44700000 밖으로 -2573.202637	40600000 44700000 밖으로 -2579.212891
44700000 51600000 나갔다 -4001.864014	44700000 51600000 나갔다 -4007.759766
.	.

4.2 이단계 탐색

<표 3>은 ECHOS에서 이단계 탐색의 성능을 보여주고 있다. HTK에서는 이단계 탐색방식을 제공하지 않기 때문에 성능 비교를 하지 못했다. 후향 탐색에서 트라이그램 적용에 의하여 상대적으로 8~16%의 오류를 감소하였다. 최종적으로 단

어간 모델을 추가한 결과 렉시컬 트리를 사용하는 베이스라인 인식기와 비교하여 5배의 인식시간 증가에 46%의 상대적 오류율 감소를 얻었다. ECHOS는 인식시간을 줄이기 위하여 세부 튜닝을 하지 않은 상태이므로 인식시간 감소의 여지가 있다.

<표 3> 이단계 탐색 방법 및 단어간 모델 적용에 따른 ECHOS의 성능

탐색 방법	단어간 모델	인식률 (%)	인식시간 (초/문장)
Forward bigram	no	75.1	3.5
Forward bigram + backward trigram	no	79.0	4.0
Forward bigram	yes	85.5	15.8
Forward bigram + backward trigram	yes	86.6	17.0

4.3 Julius와의 성능 비교

ECHOS의 최대 성능을 조사하기 위하여 가우시안의 개수를 16으로 증가하여 인식실험을 수행하고 Julius와 비교하였다. Julius는 렉시컬 트리 탐색을 지원하지만, 바이그램 전향 탐색과 동시에 트라이그램 후향 탐색을 동시에 하는 방식으로 탐색을 하고 있다. 반면에, ECHOS는 전향 탐색으로 구해진 인식결과로부터 후향으로 좀 더 정교한 모델을 사용하여 탐색한다. Julius는 가지치기할 때 제한된 노드의 개수만을 남기고 나머지는 제거하지만, ECHOS는 빔폭을 벗어나는 노드들만을 제거한다. <표 4>는 ECHOS와 Julius의 인식성능을 비교한다. 탐색방법에 차이가 있고 적용 파리미터들이 달라서 정확한 비교 분석은 어렵지만, 대체로 대등한 결과를 나타내고 있음을 볼 수 있다.

<표 4> Julius와 ECHOS의 성능 비교 (렉시컬 트리 탐색, 단어간 모델, 16개 가우시안을 갖는 연속 HMM 음향모델, 바이그램 전향 탐색, 트라이그램 후향 탐색이 적용됨).

플랫폼	가지치기 방법	언어모델 파라미터	탐색방법	인식률(%)
Julius	#상태노드: 1500	LM weight: 8.0 Insertion penalty: -2.0	Forward	87.9
			Forward + backward	93.7
ECHOS	빔폭: 125	LM weight: 5.0 Insertion penalty: 10.0	Forward	92.1
			Forward + backward	93.9

5. 결 론

교육 및 연구를 위하여 개발된 한국어 음성인식 플랫폼인 ECHOS의 구조, 기능, 구현 및 평가 결과를 소개하였다. 개발된 플랫폼은 쉽고 작으면서 한글 처리가 가능한 객체기반의 구조를 가진다. 8000단어 연속음성인식 태스크에 대하여 플

랫 렉시콘을 사용하는 공개 음성인식 플랫폼인 HTK와 성능을 비교하였다. 플랫 렉시콘을 사용할 경우, ECHOS와 HTK는 비슷한 인식률을 보이고 인식시간은 ECHOS가 4배 정도 오래 걸린다. 렉시컬 트리를 사용한 ECHOS는 플랫 렉시콘을 사용한 HTK에 비하여 인식시간을 50%로 감소하지만, 상대적으로 65%의 오류를 증가시켰다. 단어간 모델, 단일음 단어 처리, 지속시간 모델, 이단계 탐색을 적용하여 베이스라인 렉시컬 트리 탐색에 비하여 상대적으로 46%의 오류를 감소시켰다. 렉시컬 트리를 사용하는 Julius 플랫폼과 비교한 결과 대등한 수준의 인식률을 나타내었다. 이 플랫폼은 SITEC [20]에서 무료로 배포되고 있으며, 국내 음성인식 분야의 저변을 확대하고, 연구자들이 알고리듬 연구에 전념할 수 있는 토대를 마련할 것으로 기대된다.

참 고 문 헌

[1] HTK Home page. http://htk.eng.cam.ac.uk

[2] CMU Sphinx: Open Source Speech Recognition. http://www.speech.cs.cmu.edu/sphinx/Sphinx.html

[3] Automatic Speech Recognition: Software, http://www.isip.msstate.edu/projects/speech/software/

[4] Multipurpose Large Vocabulary Continuous Speech Recognition Engine Julius. http://www.ar.media.kyoto-u.ac.jp/members/ian/doc

[5] ezCSR, http://speech.chungbuk.ac.kr/~owkwon/srhome/index.html

[6] 권오욱 외, "한국어 음성인식 플랫폼의 설계", *말소리*, 제51호, 2004. 9.

[7] 권오욱 외, "한국어 음성인식 플랫폼 (ECHOS) 개발", *한국음향학회지*, 제24권 제8호, pp. 423-430, 2005.

[8] Standard Template Library Programmer's Guide, http://www.sgi.com/tech/stl/

[9] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, 1993.

[10] F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, 1999.

[11] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", *IEEE Trans. ASSP*, vol. 28, pp. 357-366, Aug. 1980.

[12] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech", *Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738-1752, 1990.

[13] Aurora, Distributed Speech Recognition, http://portal.etsi.org/stq/kta/DSR/dsr.asp

[14] M. K. Ravishankar, *Efficient Algorithm for Speech Recognition*, Ph.D. thesis, CMU, 1996.

[15] J. J. Odell, *The Use of Context in Large Vocabulary Speech Recognition*, Ph.D. thesis, Cambridge University, 1995.

[16] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing*, Prentice Hall, 2001.

[17] F. Wessel, R. Schluter, et al., "Confidence measures for large vocabulary continuous speech recognition", *IEEE Trans. Speech Audio Processing*, Vol. 9, pp 288-298. March 2001.

[18] V. Steinbiss, "Sentence-hypotheses generation in a continuous-speech recognition system", *Proc European Conf on Speech Communication and Technology*, Paris, pp. 51-54, 1989.

[19] R. Schwartz and S. Austin, "A comparison of several approximate algorithms for finding multiple (N-BEST) sentence hypotheses", *Proc. ICASSP-91*, Vol. 1, pp. 701-704, 1991.

[20] SiTEC: http://www.sitec.or.kr/index.asp

[21] S. Young, G. Evermann, et al., *The HTK Book (for HTK Version 3.3)*, 2005.

접수일자: 2006년 8월 14일
게재결정: 2006년 9월 21일

▶ 권오욱(Oh-Wook Kwon) : 교신저자
주소: 361-763 충청북도 청주시 흥덕구 개신동 12번지
소속: 충북대학교 전기전자컴퓨터공학부
전화: 043)261-3374
E-mail: owkwon@chungbuk.ac.kr

▶ 권석봉(Sukbong Kwon)
주소: 305-732 대전광역시 유성구 문지동 103-6번지
소속: 한국정보통신대학교(ICU)
전화: 042)866-6221
E-mail: sbkwon@icu.ac.kr

▶ 윤성락(Sungrack Yun)
주소: 305-701 대전광역시 유성구 구성동 373-1번지
소속: 한국과학기술원(KAIST) 전자전산학과
전화:
E-mail: yunsungrack@kaist.ac.kr

▶ 장규철(Gyucheol Jang)
주소: 305-701 대전광역시 유성구 구성동 373-1번지
소속: 한국과학기술원(KAIST) 전자전산학과
전화:
E-mail: tupp@kaist.ac.kr

▶ 김용래(Yong-Rae Kim)
주소: 361-763 충청북도 청주시 흥덕구 개신동 12번지
소속: 충북대학교 제어계측공학과
전화:
E-mail: kyr0717@chungbuk.ac.kr

▶ 김봉완(Bong-Wan Kim)
주소: 570-749 전북 익산시 신용동 344-2번지
소속: 원광대학교 음성정보기술산업지원센터(SiTEC)
전화: 063) 850-7452
E-mail: bwkim@sitec.or.kr

▶ 김회린(Hoirin Kim)
주소: 305-732 대전광역시 유성구 문지동 103-6번지
소속: 한국정보통신대학교(ICU)
전화: 042) 866-6139
E-mail: hrkim@icu.ac.kr

▶ 유창동(Changdong Yoo)
주소: 305-701 대전광역시 유성구 구성동 373-1번지
소속: 한국과학기술원(KAIST) 전자전산학과
전화: 042) 869-3470
E-mail: cdyoo@ee.kaist.ac.kr

▶ 이용주(Yong-Ju Lee)
주소: 570-749 전북 익산시 신용동 344-2번지
소속: 원광대학교 음성정보기술산업지원센터(SiTEC)
전화: 063) 850-7451
E-mail: yjlee@wonkwang.ac.kr