음성인식 플랫폼의 설계 방향

2004. 5. 8.

충북대학교
권오욱

음성인식 플랫폼 개발

목표: 한국어 연속음성인식 플랫폼 개발

- 교육 및 연구를 위한 음성인식 공통 플랫폼
- 알고리듬 비교 대상

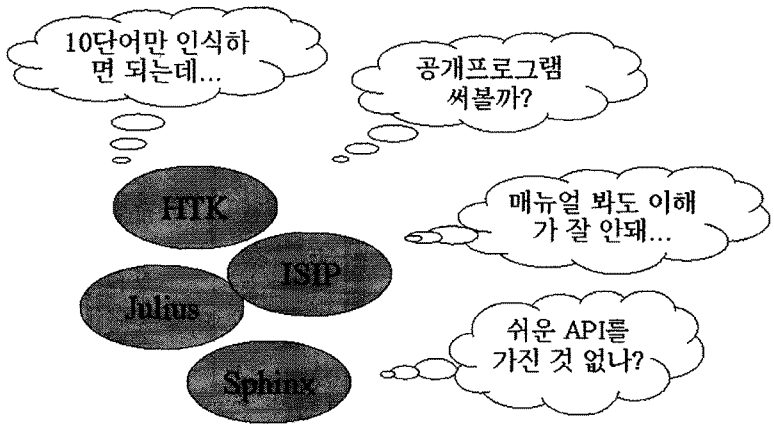
기간: 2004.5.1. – 2005.1.31. (9개월)

지원: **SITEC**

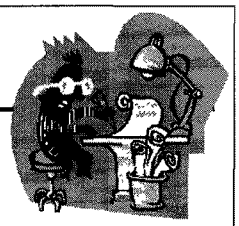
참여기관:

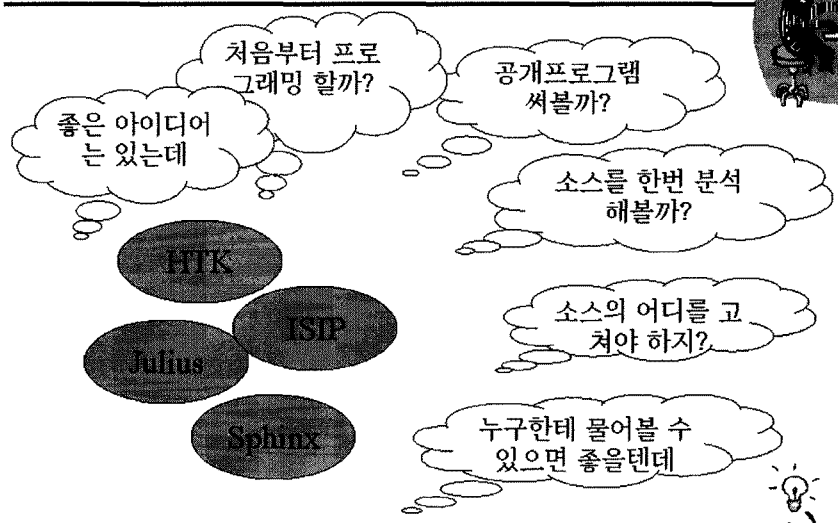
- 충북대
- ICU
- KAIST

음성인식 초보자의 고민



음성인식 연구자의 고민





필요성

교육용 플랫폼

- 기존에 공개된 음성인식기가 다수 있으나, 내부 구조 파악 어려워 독자적인 알고리듬의 신뢰성 있는 검증 어려움
- 음성인식기의 구조가 복잡하여 음성인식 연구개발 분야의 신규 진입이 어려움

연구용 플랫폼

- 내부구조에 대한 문서화가 갖추어진 공통 개발 도구는 연구역량을 핵심기술에만 집중할 수 있도록 함

공통 플랫폼

- 대학 및 연구소 별로 음성인식 연구결과를 발표하고 있으나 공통의 플랫폼을 사용하지 않아서 성능향상 결과의 검증이 어려움
- 공통 플랫폼 사용으로 새로운 알고리듬의 구현이 용이함
- 국내 공동연구 그룹간의 공통 엔진으로 사용 가능함

사례 조사(1)

Open softwares

- 음성인식 소프트웨어 공개
- 낮은 진입 장벽
- 신뢰성 있는 검증

Sphinx

- CMU Resource management task
- SourceForge에 공개

HTK

- 연구개발의 사실상의 표준 공개 소프트웨어
- 분산음성인식 시스템의 특징추출 표준화 작업(Aurora-2, Aurora-3)의 표준 인식기

사례 조사(2)

ISIP

- Mississippi 대학에서 공개한 Foundation class로 구성된 객체지향 음성인식기
- Aurora-4의 표준 인식기로 사용됨.

Julius

- 교토대 개발 시작, 최근에는 컨소시엄 형태로 일본 대학들의 대어휘 음성인식 연구개발 지원

ezCSR

- 충북대학교 객체지향 인식기

MGR

- HTK의 축소된 객체지향 버전

설계 방향(1)

교육 측면

- 쉬운 프로그램 → 알고리듬 이해에 중점
- 프로그램 설명 문서화 → 사용자 및 프로그래머 매뉴얼
- 표준 문서화 → UML 채택

연구측면

- 최근 연구동향 반영하여 필수적인 모듈 제공 → ETSI 특징추출
- 사용자 알고리듬 치환이 쉬운 구조 → 객체지향, 상속
- 파라미터 조정가능한 모듈 → 설정 화일

설계 방향(2)

소프트웨어공학 측면

- 모듈 구조(쉽게 다른 모듈로 치환 가능) → 라이브러리 공장
- 재사용이 쉬운 프로그램 → 주요 모듈에 대한 사용예제 프로그램 제공
- 일관성 있는 인터페이스 → 모든 클래스들이 비슷한 구조를 유지
- 고수준 표준 언어 사용 → 알고리듬에 주력

ECHOS

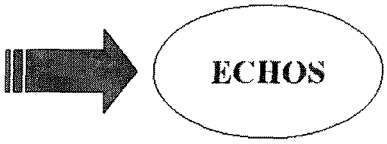
Easy
- 이해하기 쉬운 프로그램
- UML 기반 프로그램 설명 문서
- 높은 수준 API 제공

Compact
- Standard template Library (STL) 사용
- 성능 향상이 미미한 모듈 정리

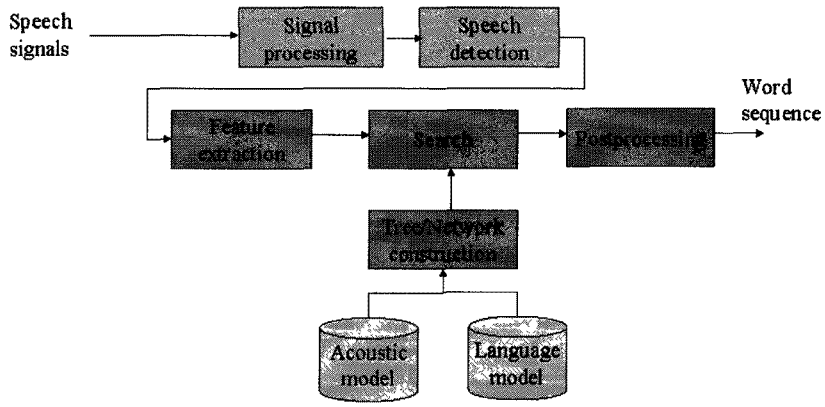
Hangeul
- 한국어 처리 모듈 보강
- 한국어 발음사전, 발음변환, 형태소분석

Object-oriented
- 모듈 구조
- 재사용 예제 제공

Speech recognizer
- 신호처리 모듈 강화
- Decoder 위주, HTK로 훈련



플랫폼 구조



입출력/성능 규격

입력

- 8/16 kHz, 16 bit PCM

출력

- 1-best
- Lattice
- Word likelihood

성능

- 연속음성인식: 30,000단어
- 핵심어 검출: 3,000단어
- 연속 청취형(continuous listening) 끝점 검출

기능(1)

신호처리 및 특징추출
- 잡음제거. Spectral subtraction*, Wiener filtering*, ETSI*
- 표준 특징 추출: MFCC*, ETSI*, PLP*
- 음성 검출: Energy-based*, ETSI-based*
- 채널 보상: ETSI*

음향 모델
- Continuous HMM*
- State sharing*
- HTK compatible*
- Decision tree*
- Semi-continuous HMM
- Online speaker adaptation

언어모델
- FSN*, Bigram*, Trigram*
- Keyword, class n-gram
- 발음사전*, 자동발음변환*, 형태소해석

탐색
- FSN search*, Search tree*
- Lattice* → N-best list*
- Two-pass search*
- 1-best forced alignment*
- Utterance verification
- Confidence measure
- Dynamic vocabulary
- Cross-word Trigram forward search
- Look-ahead Gaussian selection

기능(2)

시스템
- Windows, Linux
- Visual C++, Standard Template Library (STL)
- Mono channel, Wav/Raw format
- No SAPI support

훈련: **IITK** 이용
- Reestimation
- Decision tree-based state sharing
- Offline speaker adaptation

비교(1)

Module	ECHOS-1.0	HTK-3.2	Julius-3.4.1	Sphinx-3
Signal processing & Feature extraction	Spectral subtraction*, Wiener filtering*, ETSI* MFCC*, ETSI*, PLP* EPD: Energy-based*, ETSI-based* Channel comp.: CMS, ETSI*	MFCC, PLP, VTLN, cepstral mean & variance normalization, variance scaling Energy-based speech/silence detection	MFCC_E_D_N_Z AIFF, AU, NIST, SND and WAV(ADPCM) Spectral subtraction Remove DC offset	MFCC, PLP, CMN
Acoustic modeling	Continuous HMM* Covariance: Diag*, Full* State sharing* HTK compatible* Decision tree* Multiple pron.* Semi-continuous HMM Online speaker adaptation	Continuous/Semi-continuous/Discrete Diagonal/Full covariance Gaussian mixture HMMs Decision tree state-clustering Multiple pron., Cross-word Offline supervised SA using MLLR and MAP Online unsupervised SA using MLLR Two-model reestimation Global feature transform	HMM context dependent phoneme models (tri-phone) tied mixture and phonetic tied-mixture model Support model skip transition Support inter-word short pause handling Support binary HMM	Continuous and semi-continuous HMM Flexible feature vector Single or 4 streams Flexible HMM topology State typing with senones CART-based decision tree Multiple pronunciation
Language modeling	FSN*, Bigram*, Trigram* Class N-gram, Keyword Hangeul dictionary*, text-to-pronunciation* Morphology analysis	Lattice-based grammar format Word-pair grammar Back-off bigram n-gram tool set class n-gram	2-gram and reverse 3-gram(standard ARPA) Binary format Class N-gram	N-gram Statistical Language Modeling Toolkit

비교(2)

Module	ECHOS-1.0	HTK-3.2	Julius-3.4.1	Sphinx-3
Search algorithm	FSN Search* Search tree* Lattice*→N-best list* Two-pass search* first pass : 2-gram and tree network search* second pass : 3-gram stack decoding* 1-best forced alignment* Utterance verification Confidence measure Dynamic vocabulary	Token passing algorithm Bigram or FSN Cross-word triphone models Lattice & N-best output Forced alignment Lattice post-processing: Lattice pruning, Finding 1-best, LM expansion	Two-pass strategy first pass : 2-gram and tree network search second pass : reverse 3-gram decoding stack decoding Gaussian Pruning Confidence measure	Flat decoder (slow) Pseudo-trigram Lextree decoder (fast): Any N-gram Subvector quant. based on Gaussian selection
Prog. Lang.	C++/STL	C	C	C
Systems	Linux, Windows	Unix/Linux, Windows, Cygwin	Linux, Solaris, Digital UNIX	Linux, Unix, Windows NT
Comments	Decoder Object-oriented	De facto standard for training	Decoder, Control from client process via Network	First speech recognizer
Limitations	Cross-word, Trigram forward search, Look-ahead, Gaussian selection	Dynamic vocabulary Multi-channel, multi-thread Phoneme look-ahead Gaussian selection		

연구결과물

소프트웨어

- 음성인식 플랫폼
- 시연 인식기: 연속숫자음인식기

문서

- 사용자 매뉴얼
- 프로그래머 매뉴얼

계획

1차년도

- 기존 소프트웨어 기능 및 구조 분석(HTK, Sphinx, Mississippi, Julius)
- 요구기능 분석
- 시스템 구조 설계
- 모듈 정의
- 모듈 입출력 규격 작성
- 핵심 모듈(*표시) 구현
- 검증

2차년도(미정)

- 부가 기능 구현

추진 방안

음성인식 전문가 회의 자문
- 요구 기능
- 검증 방안
- 활용 방안
- 활성화 방안
- 공개 프로그램 제공

플랫폼 홍보
- 논문 발표

결론

ECHOS (Easy Compact Hangeul Object-oriented Speech recognizer)
- 쉬운 플랫폼
- 문서화가 잘된 플랫폼
- 모듈 구조

제안이나 코멘트를 바람

음성인식 성능 향상에 기여하기를 기대