분산 메모리 다중프로세서 환경에서의 병렬 음성인식 모델

A Parallel Speech Recognition Model on Distributed Memory Multiprocessors

정 상 화*, 김 형 순**, 박 민 욱*, 황 병 한**
(Sang Hwa Chung*, Hyung Soon Kim**, Min Uk Park*, Byung Han Hwang**)

※ 이 논문은 1996년도 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음.

요 약

본 논문에서는 음성과 자연언어의 통합처리를 위한 효과적인 병렬계산모델을 제안한다. 음소모델은 연속 Hidden Markov Model(HMM)에 기반을 둔 문맥종속형 음소를 사용하며, 언어모델은 지식베이스를 기반으로 한다. 또한 지식베이스를 구성하기 위해 계층구조의 semantic network과 병렬 marker-passing을 추론 메카니즘으로 쓰는 memory-based parsing 기술을 사용한다. 본 연구의 병렬 음성인식 알고리즘은 분산메모리 MIMD(Multiple Instruction Multiple Data) 구조의 다중 Transputer 시스템을 이용하여 구현되었다. 실험결과, 본 연구의 지식베이스 기반 음성인식 시스템의 인식률이 word network 기반 음성인식 시스템보다 높게 나타났으며 code-phoneme 통계정보를 활용하여 인식성능의 향상도 얻을 수 있었다. 또한, 성능향상도(speedup) 관련 실험들을 통하여 병렬 음성인식 시스템의 실시간 구현 가능성을 확인하였다.

ABSTRACT

This paper presents a massively parallel computational model for the efficient integration of speech and natural language understanding. The phoneme model is based on continuous Hidden Markov Model with context dependent phonemes, and the language model is based on a knowledge base approach. To construct the knowledge base, we adopt a hierarchically-structured semantic network and a memory-based parsing technique that employs parallel marker-passing as an inference mechanism. Our parallel speech recognition algorithm is implemented in a multi-Transputer system using distributed-memory MIMD multiprocessors.

Experimental results show that the parallel speech recognition system performs better in recognition accuracy than a word network-based speech recognition system. The recognition accuracy is further improved by applying code-phoneme statistics. Besides, speedup experiments demonstrate the possibility of constructing a realtime parallel speech recognition system.

I. 서 론

현대 사회에서 인간은 다양한 매체를 통하여 정보를 받아들이고 있으며, 이에 따라 보다 편리하고 신속한 정보 교환을 위한 인간과 기계사이의 인터페이스 기술이 중요하게 대두되고 있다 [1]. 음성인식에 있어서의 중요한 연구중의 하나는 음성과 자연언어의 통합처리인데, 이 방법은 word-level의 지식뿐만 아니라 syntax 등의 high-level 정보를 사용하여 다수의 가설을 해결하고 있다. 그러나 이 통합처리 기법은 여러 레벨의 지식원들을 통한 대규모 계산이 요구되므로 광범위한 domain을 위해 구성된 실질적인 지식베이스(knowledge base)상에서는 계산속도가 심각하게 저하될 수 있다.

본 논문에서는 phonetic, lexical, syntactic layer 등의 지식원을 효과적으로 결합할 수 있는 계층구조(hierarchically-structured)의 지식베이스를 구축하고, 병렬 marker-passing을 추론 메카니즘으로 쓰는 memory-based parsing 기술을 사용하여 음성과 자연언어의 통합처리를 위한 병렬계산모델을 개발하였다. Speech front-end(SF) 모듈에서의 음소 모델은 한국어 음소 변이음(allophone) 기반의 HMM을 사용하였다. 또한 본문에서는 음성인식 과정에서 발생하는 insertion, deletion, substitution, word boundary detection 등의 speech-specific problem을 분석하고 병렬처리에 의한 해를 제공한다. 본 연구의 병렬 음성인식 알고리즘은 분산메모리 MIMD 구조의 다중 Transputer 시스

* 부산대학교 컴퓨터공학과
** 부산대학교 전자공학과
접수일자 : 1999년 2월 24일

템을 이용하여 구현되었다. 실험을 통하여 음성인식 시스템의 병렬화에 의한 실시간 음성인식의 가능성과 병렬 음성인식 모델을 효과적으로 구성할 수 있는 적정 규모의 분산메모리 다중 처리 시스템을 보여준다.

본 논문의 구성은 다음과 같다. 2장에서는 음성과 자연언어 통합처리와 병렬음성인식에 관한 관련 연구를 기술하며, 3장에서는 분산메모리 다중프로세서 환경에서의 병렬 음성인식시스템을 제시한다. 4장에서는 음성처리에서 발생되는 특성을 이용한 시스템 성능향상에 대하여 기술한다. 5장에서는 실험을 통하여 병렬 음성인식시스템의 성능을 평가하고, 6장에서는 결론을 맺는다.

II. 관련연구

외국의 경우, 다양한 지식원을 활용하여 음성과 자연언어를 통합처리하는 연구가 꾸준히 진행되어 왔다. Paeseler [12]는 chart parsing 알고리즘을 음성 인식 연구에 적용하였으며, Hayes [4]는 spoken language의 parsing을 위해 semantic case frame을 사용하였다. Young [17]은 pragmatic knowledge를 활용하여 사용자의 다음 발언에 대한 동적인 expectation을 가능하게 하는 MINDS라는 시스템을 개발하였다. Kitano [7]는 memory-based parsing을 일본어와 영어간의 음성 자동 번역 시스템을 위하여 사용하였다.

한편 병렬 parsing에 관한 연구를 살펴보면, 대부분의 연구가 spoken language에 비해 구현하기가 쉬운 written language에 집중되었다. Huang과 Guthrie [6]는 syntax와 semantics의 통합처리에 기본을 둔 자연어 parsing을 위한 병렬 모델을 제안하였으며, Waltz와 Pollack [16]은 connectionist 모델을 적용한 MPP 모델을 개발하였다. Spoken language에 관한 연구로는 Giachin과 Rullent [3]가 Transputer-based distributed architecture상에 개발한 병렬 parser가 있다. 또한 Chung과 Moldovan [2]은 AI 응용을 위하여 개발된 SNAP MPP 시스템상에 음성과 자연어의 통합처리를 위한 병렬 parser를 개발하였다. HMM 기반 병렬 음성인식 시스템에 관한 연구로써, Gijsbert Huijsen [5]은 discrete HMM을 기반으로 하여 nCube2 machine에서 모델링과 음성인식을 시도함으로써 병렬화를 통한 음성인식의 모델링과 인식에 있어서의 시간 단축을 보여주었으며, AT&T [14,15]에서는 인식하려는 domain내의 모든 문장을 multi-threading하여 처리하는 음성인식 시스템을 제시하였다.

III. 병렬 음성인식 시스템

3.1 시스템 개요

본 병렬 음성인식 시스템은 그림 1에 제시된 바와 같이 speech front-end (SF) 모듈과 natural language understanding (NLU) 모듈, speech understanding (SU) 모듈 및 knowledge base (KB)로 나누어진다. SF 모듈은 화자독립 연속음성을 처리하여 phonetic code stream을 SU 모듈에 제공해준다. SU 모듈은 SF 모듈에 의하여 공급된 phonetic code를 이용하여 matching phoneme sequence를 찾아낸다. NLU 모듈은 high-level 정보를 이용하여 word candidate를 예측하여 SU 모듈이 다수의 가설을 효과적으로 처리하게 한다. NLU 모듈은 SU 모듈에 의해 인식된 word candidate들을 사용하여 meaning representation을 구축하고 이를 토대로 sentence output을 형성한다.

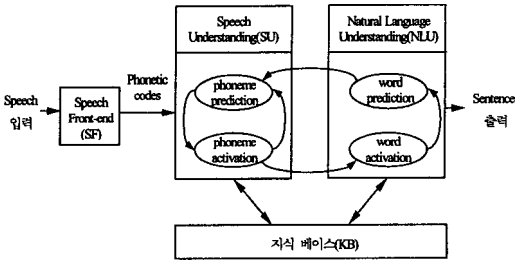


그림 1. 병렬 음성인식 시스템
Fig. 1. A parallel speech recognition system.

3.2 SF 모듈

본 논문에서의 SF 모듈의 구성도가 그림 2에 나타나 있다. SF 모듈의 첫 단계인 음성특징분석의 목적은 언어정보, 즉, 음소들간의 차이에 해당하는 음향학적 특성에는 민감하면서도 그 이외의 음향적 변화(배경잡음, 채널왜곡, 화자 차이, 발음 태도 등)에는 둔감한 음성특징 파라메터들을 추출하는 것이다. 본 논문에서는 일반적으로 음성인식에 널리 사용되는 특징 파라메터인 mel-frequency cepstral coefficients (MFCC) [10,11]를 사용하였다. MFCC는 음성 신호를 주파수 영역으로 변환한 다음, 청각기관의 주파수 선택특성을 고려한 대역 필터군 출력을 구하고, 이를 log scale에서 Inverse Discrete Cosine Transform (IDCT)하여 구한다. SF 모듈에서 사용한 특징 파라메터 차수로는 지금까지 설명한 MFCC(12차)와 음성의 에너지, 그리고 그 미분치들을 합하여 모두 26차를 사용하였다.

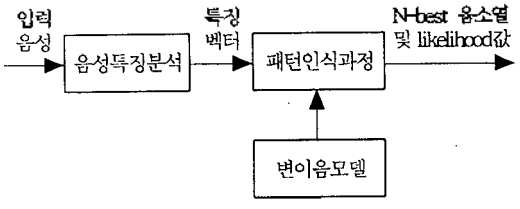


그림 2. Speech front-end(SF) 모듈
Fig. 2. A speech front-end(SF) module.

패턴인식과정은 음성특징분석과정에서 추출된 음성특징 계수들과 가장 잘 부합되는 언어적 표현을 찾아내는 과정이다. 이를 위해 먼저 패턴인식을 하기 위한 음성의 기본 단위(단어, 반음절, 음소, 변이음 등)를 정한 다음, 훈련용 음성 데이타로부터 미리 이들 음성단위에 해당하는

각각의 대표패턴 또는 모델을 구해서 저장한다. 그 다음으로 인식하고자 하는 입력음성의 특징패턴이 분석되면 이를 저장된 대표패턴 또는 모델들과 비교하여 가장 가까운 패턴들에 해당하는 음성단위들을 인식된 단어 또는 음소의 후보로 결정하게 된다. 본 논문의 SF모듈에서는 HMM [13]이라는 통계적 방법을 사용하였으며, 이 방법은 음성 단위에 해당하는 패턴들의 통계적인 정보를 확률모델 형태로 저장하고 미지의 입력패턴이 들어오면 각각의 모델에서 이 패턴이 나올 수 있는 확률을 계산함으로써 이 패턴에 가장 적합한 음성단위를 찾아내는 방법이다. 이 방법에서는 음성신호로부터 모델의 파라메터들을 추정하고 추정된 모델과 입력된 음성패턴과의 유사도를 측정하는 과정들이 명확하게 정의되어 있으며, 모델훈련에 필요한 양의 음성 데이터가 준비될 경우 성능 면에서도 가장 우수한 것으로 평가되고 있어서 현재 음성인식을 위한 패턴인식 방법으로 가장 널리 사용되고 있다

본 논문의 SF모듈에서는 패턴매칭을 위한 음성 기본단위로 46개의 변이음을 선정하였다. 변이음은 음소가 그 실현되는 환경의 차이 및 기타 이유로 인해 나타나는 다양한 형태를 고려한 음소를 말한다. 그리고, 이들 각각의 음소모델들은 연속 HMM(continuous density HMM) 기반의 문맥독립형(context-independent) 음소 모델인 monophone과 문맥종속형(context-dependent) 음소 모델인 triphone 모델의 두가지 형태로 모델링하였다. 문맥독립형 음소 모델의 경우 관찰빈도가 상대적으로 높아서 훈련하기는 용이하나 인접음소의 영향에 따른 세밀한 변화특성의 묘사에 취약하고, 이에 비해 문맥종속형 음소 모델은 구체적인 환경하에서의 음소특성을 표현하는 데에는 유리하나 제한된 데이터로부터 수많은 모델들을 신뢰성 있게 모델링하는 데에 한계가 따른다. 본 논문에서는 SF모듈의 인식실험을 위해 Entropic사의 Hidden Markov Model(HMM) 기반의 음성인식 Tool인 HTK(HMM Tool Kit) V2.0을 사용하였다.

3.3 Alignment scoring 모델

어휘에 대한 정보를 전혀 이용하지 않을 경우, 현재 기술 수준에서의 불특정화자 변이음 인식 성능은 최고 70% 대에 불과하다 [8]. 따라서, SF 모듈의 결과로 얻어지는 후보 변이음 열에는 다수의 인식오류가 포함된다. 이들 인식 오류들은 insertion, deletion, substitution들로 나누어 질 수 있다. Insertion은 SF 모듈이 사용자의 부적절한 발음 등의 영향으로 over-segmentation하여 인식결과에 다른 변이음이 첨가된 것을 말하며, deletion은 반대로 under-segmentation되어 필요한 변이음 정보가 상실되는 것을 말한다. 그리고 substitution은 어떤 인식대상 변이음이 다른 변이음으로 대체된 경우를 말한다. 논문에서는 각각의 변이음들이 insertion, deletion, 또는 다른 변이음으로 substitution될 가능성들을 통계 자료로부터 분석하고, 이 정보를 이용하여 병렬 처리 기반의 SU 모듈이 insertion, deletion, substitution의 문제를 해결할 수 있는 근거를 제공하고자 한다. 이를 위해 Dynamic Time Warping(DTW) 알고리즘을 이용하여 reference 문장과 인식된 문장간의 각 음소들의 insertion, deletion, substitution의 통계 데이터를 구하였다.

본 논문에서는 음소들간의 더 정확한 insertion, deletion, substitution의 통계적 분포를 유도하기위해 reference 문장의 음소 분할된 시간정보를 이용하였다. 우선 관측된 음성의 특징 파라메터를 Viterbi decoding하여 훈련된 모델에서 발견될 최적의 state열을 구한다.

3.4 지식베이스

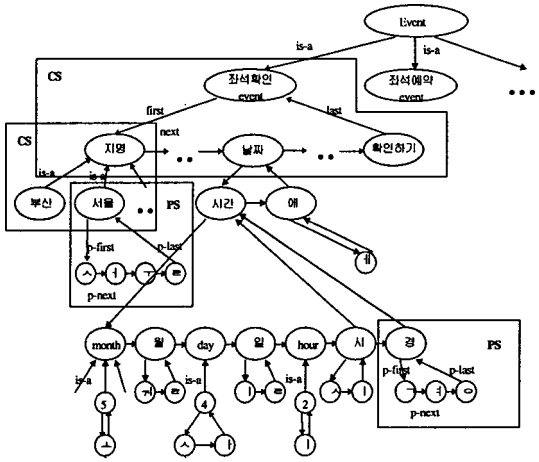


그림 3. 계층구조의 지식베이스 예
Fig. 3. An example of hierarchical knowledge base.

본 연구에서는 여러 계층의 지식원들사이의 밀접한 상호연관성을 지원하기 위하여 계층구조의 지식베이스를 사용하였다. 이러한 계층구조하에서 memory-based parsing을 수행하기 위하여 concept sequence(CS)에 기반을 둔 building block을 구성하였다. CS는 하나의 concept sequence root(CSR)와 하나 이상의 concept sequence element(CSE) 노드로 구성되어 있다. 이는 문장과 어절을 이루는 단위가 되며, CSE 노드는 first, next, last link로 연결되어 있다. 이와 유사하게 phoneme sequence(PS)는 word를 이루는 단위가 되는 것으로, 하나의 concept node에 부착되어 p-first, p-next, p-last link로 연결되어 있다. 해당하는 concept node에 부착된 각 PS는 계층구조의 가장 하위 계층을 형성한다.

그림 3은 본 연구에서 사용하고 있는 컴퓨터 비서 에이전트 domain의 지식베이스 일부를 예로서 보여주고 있다. 여기서 좌석확인 event는 CS root 노드이며 지명, 날짜, 확인하기 등은 이에 대응되는 CSE 노드들이다. 또한 그림에서 볼 수 있듯이 이들 CSE 노드는 하위 레벨의 CSE를 가질 수 있으며 CSE 노드에는 대응되는 PS들이 배치되어 있다.

3.5 Marker-passing

본 연구에서는 문장인식을 위하여 marker-passing을 통한

memory-based parsing 기술을 사용하였다. 이는 marker-passing을 이용하여 parallelism을 최대한 활용하는 추론 메카니즘을 구현하는 것으로, parsing의 진행 과정은 다음과 같다. 우선 지식베이스를 이루는 지식원(단어, 어절)들은 위에서 보여준 바와 같이 CS로 memory에 저장된다. Parsing 과정은 기본적으로 top-down prediction과 bottom-up activation을 통해 이루어진다. Parsing이 시작되면 지식베이스의 상위 계층을 구성하는 node로부터 하위 계층으로 prediction이 전파된다. 처음에는 모든 CSR과 그것이 포함하는 하위 계층의 첫 번째 CSE와 PSE가 가설로서 prediction되지만, phoneme candidate가 root 프로세서로부터 들어오게 되면, prediction된 node들 중 그 input과 일치하는 node만 activation된다. 한 node에서의 prediction과 activation의 충돌은 node에 연결된 link를 통해 다음 node로 prediction이 전파되게 한다. 초기에는 모든 문장이 가설로서 prediction되어도 parsing이 진행됨에 따라 실제 activation이 일어나는 가설에는 한계가 있으므로 가설의 범위는 점차 줄어든다. 이러한 과정을 거치면서 CS가 인식되면 이로부터 출력문장을 generate한다.

본 연구에서 병렬 marker-passing을 위해 사용하는 marker에는 다음과 같은 종류가 있다.

- P-marker (prediction) : Concept sequence나 phoneme sequence에서 다음에 activation될 수 있는 node
- A-marker (activation) : Activation이 일어난 node
- I-marker (instance) : Activation된 node(CSE)에 대한 instance
- C-marker (cancel) : 일정 시점에서 점수가 낮은 path에 대해 역방향으로 가설을 취소

P-marker와 A-marker는 실제로 memory-based parsing을 수행하는 역할을 담당한다. P-marker는 각 가설이 지나온 경로에 대한 score를 계산하고 운반하며, A-marker는 각 phoneme의 정보를 메모리에 적재된 지식베이스에 전달하는 역할을 한다. I-marker는 인식된 단어에 대한 instance를 가리키기 위해 사용된다. Parsing이 진행됨에 따라 인식된 단어의 instance들은 I-marker사이의 일련의 link를 통하여 instance list를 형성하는데, 각 가설은 하나의 instance list를 가진다. 최종적으로 도착한 가설들에 대해 지나온 path에 대한 score를 비교한 후 선택된 가설을 backtracking하여 인식된 문장을 생성한다. Backtracking은 I-marker의 link를 사용하여 instance list를 역방향으로 추적하는 것으로 이를 통해 해당 문장을 수집할 수 있다. C-marker는 잘못된 alignment로 인해 가설에 대한 score가 한계치보다 낮은 경우 해당 가설을 제거하는 역할을 한다.

3.6 병렬 음성인식 알고리즘

본 논문의 제안하는 병렬 음성인식 알고리즘은 그림 4와 같다. 음소모듈을 통해 다수개의 후보 음소열이 생성되면, root 프로세서는 병렬 음성인식을 위한 지식베이스를 병렬 프로세서에게 전송하여 분산 적재시킨다. Root 프로세서는 후보해의 phonetic code를 음소인식 시간을 기준으로 정렬하여 병렬 프로세서에 전송하고, 각 프로세서에는 적재된 지식베이스를 이용하여 병렬 marker-passing을 수행한다. 병렬 프로세서는 root 프로세서로 결과를 전달하고, root 프로세서는 이 값을 받아 후보해중에서 가장 점수가 높은 것을 선택한다. Root 프로세서는 선택된 문장을 구성하는 단어를 인식하기 위해 병렬 프로세서로 메시지를 전송한다. Root 프로세서는 병렬 프로세서에서 전송된 결과를 이용하여 인식된 문장을 출력한다.

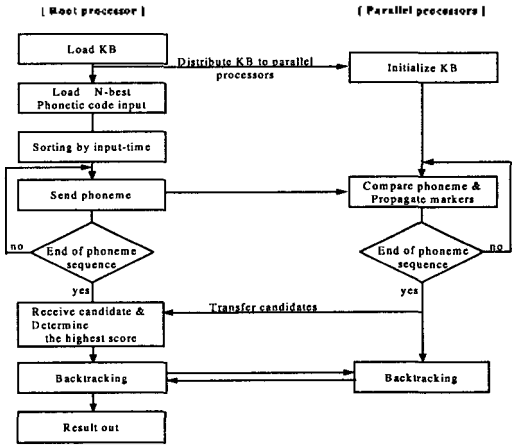


그림 4. 병렬 음성인식 알고리즘
Fig. 4. A parallel speech recognition algorithm.

3.7 병렬 machine

본 연구에서 사용한 병렬 음성인식 machine은 호스트 컴퓨터와 병렬 컴퓨터로 구성된다. 호스트 컴퓨터는 음성인식에 사용되는 입력 데이타를 공급하고 전체 인식 결과를 사용자에게 제공하며 전체 시스템을 관리하는 역할을 담당한다. 병렬 컴퓨터는 호스트로부터 들어오는 입력에 대한 다수의 후보해를 동시에 평가한다. 본 연구에서는 이상과 같은 시스템을 구성하기 위해 분산 메모리 MIMD 구조의 병렬 컴퓨터로써 가격 대 성능비가 우수하고 병렬 프로그램 개발환경이 뛰어난 다중 Transputer 시스템을 사용한다. 다중 Transputer는 1개의 root 프로세서와 16개의 processing element (PE)로 구성되어 있다. 여기서 root 프로세서는 host와 PE들간의 통신을 담당하고, 각 PE는 실제로 local control에 의해 parsing을 수행하는 역할을 한다. 각 프로세서는 Inmos사의 T805 32bit Transputer로 병렬 컴퓨터를 위해 개발된 CPU이다. T805는 다른 프로세서와 통신의 지원하기 위해 4개의 양방향 interprocessor communication link를 가지고 있으며 인접 프로세서와 20Mbits/sec로 background 통신이 가능하다. Interconnection network은 2-dimensional mesh with wrap-around를 사용한다.

VI. Speech-specific problem의 해결

4.1 Windowing

위에서 설명한 marker-parsing에서 중요하게 고려되어

야 할 부분은 음성인식 과정에서 생기는 insertion, deletion, substitution 등의 speech-specific problem이다. SF 모듈이 제공하는 phoneme sequence는 이러한 문제점을 내포하고 있으므로 지식베이스상의 phoneme과 정확한 alignment가 어렵다. 본 논문에서는 다음과 같이 prediction window와 input window를 사용하여 phonetic code stream에 나타나는 다양한 insertion, deletion, substitution에 대해 좀 더 신뢰성있는 판단을 제공한다.

Prediction window

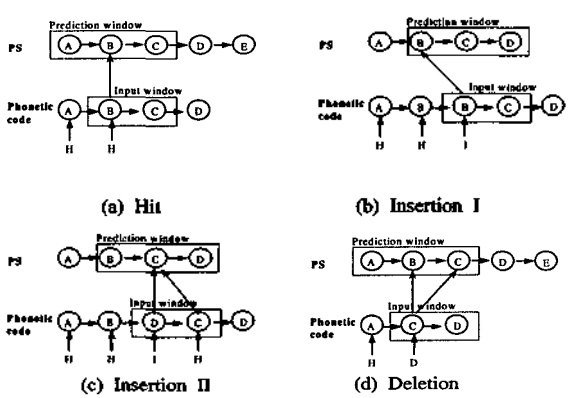
Prediction window는 parsing의 진행 과정에서 activation 가능범위를 확대하기 위해 P-marker가 가리키는 phoneme list이다. Prediction window는 현재 prediction된 node와 그 앞뒤로 이어진 node로 구성된다.

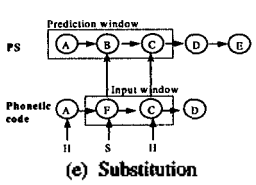
Input window

Input window는 현재 및 다음에 들어오는 phonetic code로 이루어진 list이다. SF 모듈에서 들어오는 현재의 phoneme candidate 값만으로는 insertion, deletion, substitution 여부를 정확하게 판단하기 어렵기 때문에 다음에 들어오는 phoneme candidate를 활용하여 align을 시도한다.

4.2 Window를 이용한 alignment

다음에 보여주는 그림 5는 window를 사용하여 PS와 phonetic code간의 alignment 과정을 보여준다. 그림 5(a)는 입력된 phonetic code가 prediction된 phoneme과 일치하는 정상적인 alignment를 보여준다. 그림 5(b)의 Insertion I은 입력으로 들어온 phonetic code가 이미 activation된 phoneme과 일치하는 경우이고 그림 5(c)의 insertion II는 다음에 들어올 phonetic code와 현재 prediction된 phoneme이 일치하는 경우이다. Insertion II의 경우에는 다음에 생길 hit을 미리 예측하여 align한다. 그림 5(d)의 Deletion은 현재 들어온 phonetic code가 다음에 prediction될 phoneme과 일치하는 경우이다. 마지막으로 Substitution은 다음에 들어올 phonetic code와 다음에 prediction될 phoneme이 서로 일치하는 경우로 Insertion II의 경우와 마찬가지로 다음 align시에 hit이 발생할 것을 예측할 수 있다.





H : hit
I : insertion
D : deletion

그림 5. Window상에서의 PS와 phonetic code간의 alignment
Fig. 5. Alignment between PS and phonetic code in window.

앞서 설명한 PS와 phonetic code간의 alignment 과정에서 P-marker의 이동은 다음과 같다.

- Hit : P-marker는 정상적으로 한 단계 앞으로 이동한다.
- Insertion I & II : P-marker는 이동하지 않는다.
- Deletion : P-marker는 두 단계 앞으로 이동한다.
- Substitution : Hit과 마찬가지로 한 단계 이동한다.

4.3 Code-phoneme 통계정보의 활용

본 연구에서는 각 음소의 insertion, deletion, substitution 발생빈도를 이용하여 window의 alignment시에 insertion, deletion, substitution의 신뢰성 있는 판단을 유도한다. 구체적으로, code-phoneme 통계정보는 parsing과정에서의 score 계산시 phonetic code가 가지는 점수의 반영 정도를 조절하는 기준으로 이용된다. 본 연구에서 생성한 code-phoneme 통계정보는 1) 모든 음소의 발생개수와 insertion, deletion, substitution이 일어나는 개수, 2) substitution시 각 음소가 어떤 음소로 대체 가능한지에 대한 정보, 3) 지식베이스상 음소가 대체될 수 있는 음소의 종류 및 발생 개수이다.

4.4 Code-phoneme 통계정보에 기반한 scoring

Windowing과 code-phoneme 통계정보를 이용하여 지식베이스상의 phoneme과 phonetic code사이의 alignment가 결정되고, 그 결정에 따라 문장의 점수에 현재 phonetic code가 가지는 점수를 더하는 방법이 달라진다. Hit인 경우에는 SF모듈을 통해서 입력된 phonetic code의 점수를 그대로 더하고, insertion, deletion, substitution이 발생한 경우 penalty를 부여한다. 결과로써, 지식베이스의 음소정보와 비교하여 insertion, deletion, substitution이 많은 후보해인 경우 인식종료시에 수반되는 다수개의 후보해중에서 제외될 확률을 높인다.

V. 실험 및 결과

5.1 Task Domain 및 음성 데이타베이스

실험에 사용된 음성 데이타베이스는 ETRI의 컴퓨터 비서 에이전트 시스템용 문장 목록 [9,18]을 clean 환경에서 110명의 남성 65명, 여성 45명이 발성한 음성을 16 KHz로 샘플링하여 헤더가 없는 16bit의 PCM 데이타로 저장되어 있다. 음소 모델의 훈련과 인식실험을 위해 남성화자만을 사용하였다. 훈련에는 59명이 77문장을 발음

한 총 4543개의 문장을 사용하였고, 인식 실험을 위해 나머지 6명이 77문장을 발음한 총 462문장을 사용하였다.

5.2 SF 모듈의 성능평가

본 논문에서 사용한 HMM 모델 구조는 3개의 state를 사용하고 state간의 skip을 허용 안한 경우와 6개의 state를 사용하고 state간의 skip을 허용한 경우의 두 가지를 사용하였다. 또한 각각의 경우에 대해 state당 mixture를 1, 3, 5, 6개를 가지는 연속확률분포 HMM으로 모델링하였다.

SF 모듈에서의 문법 구조는 NULL grammar를 이용하였고, 이와 별도로 우리말의 음절구성방법을 고려한 자음+모음, 자음+모음+자음, 모음+자음, 모음 형태의 순서를 이용하는 grammar도 사용하였다.

그리고 N-best의 인식결과는 lattice 구조의 형태로 병렬처리를 위한 다음단계에 넘겨진다. Lattice구조의 예는 그림 6에 나타나 있다. Lattice 구조는 음소의 시작점/끝점을 나타내는 node와 인접한 node사이에 하나의 변이음에 해당하는 link가 있고, 각 link는 그 구간에서의 해당 음소모델의 log 확률값을 가진다. Log 확률값은 음수이며 0에 가까울수록 높은 확률값을 나타낸다.

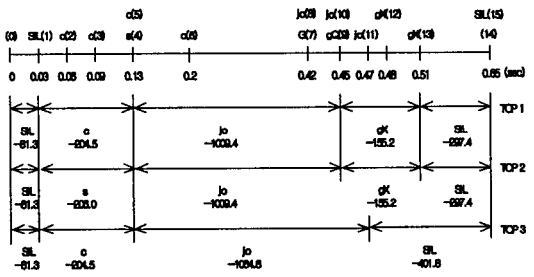


그림 6. Lattice 구조로 표현한 N-best변이음 인식결과의 예
Fig. 6. An example of N-best recognition result represented as lattice structure.

SF 모듈의 인식성능 평가수단인 %Correct와 %Accuracy는 다음과 같은 수식으로 주어진다.

$$\%Correct = \frac{N-S-D}{N} \times 100 \quad (1)$$

$$\%Correct = \frac{N-S-D-I}{N} \times 100 \quad (2)$$

여기서 N은 전체 변이음 개수, D는 삭제(deletion)된 변이음 개수, S는 대치(substitution)된 변이음 개수, 그리고 I는 삽입(insertion)된 변이음 개수를 의미한다. 인식결과, 인접음소들의 영향을 고려한 triphone이 84.9%의 %Correct와 58.7%의 %Accuracy 결과를 나타내었으며, 이는 monophone보다 10-20%의 우수한 성능을 나타낸다. 이러한 결과를 토대로 본 논문에서의 SF 모듈을 최종적으로 state 3개, mixture 6개를 가지는 triphone을 기반으로 하여 음소모델을 훈련하였고, 이 음소모델을 사용하여 18개 문장규칙을 사용하여 연속음성 인식기의 형태로 구성하였다.

5.3 병렬 음성인식 시스템의 인식률

표 1. 음성인식시스템의 인식률 비교
Table 1. Comparison between speech recognition systems.

음성인식 방법	인식률
Finite State Grammar Network을 이용한 음성인식	82.2%
지식베이스를 이용한 병렬음성인식	87.4%

본 실험에서는 기존의 finite state grammar network을 이용한 음성인식과 본 연구에서 제시하는 병렬 음성인식 시스템의 인식률을 비교하였다. 표 1에서 지식베이스를 이용한 시스템이 finite state grammar network을 이용하는 시스템보다 높은 인식성능을 가지고 있음을 보여준다. 지식베이스를 이용한 인식시스템은 생성가능한 모든 문장을 대상으로 인식이 진행되므로 더욱 향상된 인식률을 보여준다. 또한, 본 연구의 도메인에서 발생하는 insertion, deletion, substitution 정보를 penalty값으로 이용함으로써 지식베이스를 이용한 시스템의 인식 성능을 좀 더 높일 수 있었다.

본 연구에서는 hit의 penalty, deletion의 penalty, substitution의 penalty를 각각 변화시키면서 대략적인 penalty의 범위를 결정한 다음, 이 범위내에서 3가지 penalty값을 함께 변화시켜 가장 좋은 문장 인식률을 가질때의 penalty 값을 확정하였다.

5.4 성능향상도

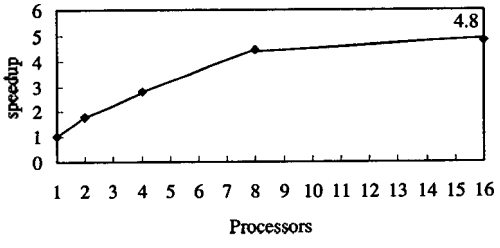


그림 7. 성능향상도
Fig. 7. Speedup.

Transputer 상에서 프로세서 수의 증가에 따른 수행속도의 향상은 그림 7과 같다. 그림 7은 프로세서를 한 개 사용했을 때의 수행시간을 1로 두었을 때 프로세서 수가 증가함에 따른 성능향상도를 보여주고 있으며, 16개의 프로세서를 사용하였을 때 4.8배의 향상을 보인다. 프로세서 수가 8개에서 16개로 증가됨에도 불구하고 성능향상도가

거의 직선에 가까운 것은 지식베이스의 규모가 프로세서 수에 비해 충분히 크지 않음에 따른 것으로 분석된다.

5.6 지식베이스 분산적재에 따른 성능향상도

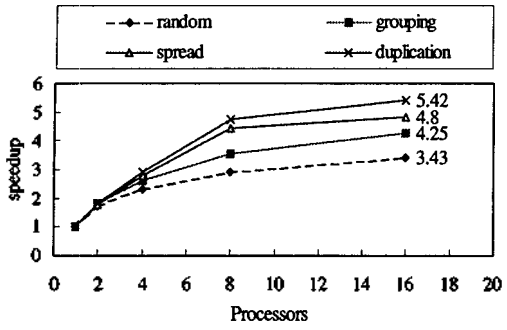


그림 8. 분산 적재 방법에 따른 성능향상도
Fig. 8. Speedup according to various knowledge base loading schemes.

그림 8에서는 각 프로세서에 분산 적재되는 지식베이스를 달리함으로써 나타나는 수행시간의 변화를 보여준다. 각 경우는 지식베이스를 random하게 적재시켰을 때(random), word group을 한 node에 적재시켰을 때(grouping), word group 내의 각 word를 인접한 다른 node에 적재시켰을 때(spread), 그리고 자주 사용되는 word group을 모든 node에 duplicate시켰을 때(duplication)를 나타낸다. 그림에서 보는 바와 같이 프로세서를 16개 사용하면 duplication의 경우가 random하게 적재한 경우보다 1.58의 성능향상을 보여준다. 또한 한 node에 같은 word group의 단어를 모두 적재시켰을 때보다 word group의 각 단어를 분리시키는 것이 더 좋은 성능을 보인다.

VI. 결 론

본 논문에서는 음성과 자연어의 통합처리를 위한 병렬 음성인식 알고리즘을 제시하였다. 음소모델은 continuous HMM을 사용하였고 언어모델은 지식베이스에 기반을 둔 계층적 semantic network을 기반으로 하였다. 또한 음성인식 과정에서 발생하는 speech-specific problem을 분석하고 병렬 음성인식 모델을 효과적으로 구현할 수 있는 분산메모리 다중처리 시스템을 개발하고 그 성능을 분석하였다. 실험결과, 본 연구의 지식베이스 기반 음성인식 시스템의 인식률이 word network 기반 음성인식 시스템보다 높게 나타났으며 code-phoneme 통계정보를 활용하여 인식성능의 향상도 얻을 수 있었다. 또한, 성능향상도 관련 실험들을 통하여 병렬 음성인식 시스템의 실시간 구현 가능성을 확인하였다. 앞으로의 연구 방향은 HMM 기반 병렬 SF모듈의 개발과 SF 모듈과 SU, NLU 모듈의 밀결합에 의한 성능향상 방안을 개발하는 것이며, 대용량 음성 DB를 사용하기 위한 knowledge representation 기술의 개발도 필요하다.

참 고 문 헌

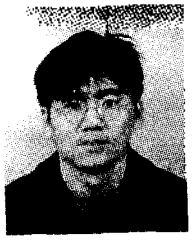
1. A. I. Rudnicky, A. G. Hauptmann, *Survey of Current Speech Technology*, CMU, Pettsburgh PA, June 4. 1993.
2. S. H. Chung, D. I. Moldovan, and R. F. DeMara, "A Parallel Computational Model for Integrated Speech and Natural Language Understanding," *IEEE Transactions on Computers*, vol. 42, no. 10, pp. 1171-1183, 1993.
3. E. P. Giachin and C. Rullent, "A Parallel Parser for Spoken Natural Language," *Proceedings of IJCAI*, pp. 1537-1542, 1989.
4. P. J. Hayes, A. G. Hauptmann, J. G. Carbonell, and M. Tomita, "Parsing Spoken Language: a Semantic Caseframe Approach," *Proceedings of COLING-86*, pp. 587-592, 1986.
5. G. Huijsen, *Parallel Implementation of Hidden Markov Models on the nCUBE2*, M. Sc. thesis, Alparon report, nr. 96-03, Delft University of Technology, 1996.
6. X. Huang and L. Guthrie, "Parsing in Parallel," *Proceedings of COLING-86*, pp. 140-145, 1986.
7. H. Kitano, "ΦDM-Dialog: A Speech-to-speech Dialogue Translation System," *Machine Translation*, 5, pp. 301-338, 1990.
8. K. F. Lee, F. Alleva, "Continuous Speech Recognition," in *Advances in Speech Signal Processing*, pp. 623-650, Marcel Dekker, New York, 1992.
9. Y. J. Lee, Design and Construction of Korean Speech Database, Final report, ETRI, 1996.
10. J. D. Markel, A. H. Gary, Jr., *Linear Prediction of Speech*, Springer-Verlag, 1776.
11. S. Davis, P. Mermelstein, "Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences," in *IEEE Trans. ASSP*, pp357-266, 1980.
12. A. Paeseler, "Modification of Earley's Algorithm for Speech Understanding," *Recent Advances in Speech Understanding and Dialog Systems*, Springer-Verlag, Berlin, 1987.
13. L. R. Rabiner, Biing-Hwang Juang, *Fundamentals of speech recognition*, 1993.
14. M. D. Riley, A. Ljolje, D. Hindle, and F. Pereira, "The AT&T 60,000 word speech-to-text system." In Proceedings of *EUROSPEECH-95*, pages 207-210, 1995.
15. Steven Phillips, Anne Rogers. "Parallel Speech Recognition," In *Preceedings of EUROSPEECH- 97*, pages 135-138, 1997.
16. D. L. Waltz and J. B. Pollack, "Massively Parallel Parsing: A Strong Interactive Model of Natural Language Interpretation," *Cognitive Science* 9, pp. 51-74, 1985.
17. S. R. Young et al., "High Level Knowledge Sources in Usable Speech Recognition Systems," *Communications of ACM*, vol. 32, no. 2, pp. 183-193, 1989.
18. 이용주, 음성 데이타베이스 설계 및 제작, 용역결과보고서, 한국전자통신연구소, 1996년 8월.

▲정 상 화(Sand Hwa Chung)

1985년 : 서울대학교 전기공학과(공학사)
1988년 : Iowa State University 컴퓨터공학(공학석사)
1993년 : University of Southern California 컴퓨터공학(공학박사)
1994년 : University of Central Florida 미국조교수(병렬처리)
현재 : 부산대학교 컴퓨터공학과 조교수
※주관심분야 : 병렬처리, 음성처리

▲김 형 순(Hyung Soon Kim)
현재 : 부산대학교 전자공학과 부교수
한국음향학회지 제17권 3호 참조

▲박 민 욱(Min Uk Park)



1996년 : 부산대학교 영어영문학과 (학사)
1998년 : 부산대학교 인지과학협동과정 (석사)
현재 : 부산대학교 컴퓨터공학 (박사과정)
※주관심분야 : 병렬처리, 음성처리

▲황 병 한(Byung Han Hwang)

1994년 : 부산대학교 전자공학과(학사)
1999년 : 부산대학교 전자공학과 (석사)
현재 : 삼성반도체연구소 TD
※주관심분야 : 음성처리, 신호처리