분산 메모리 다중프로세서 상에서의 병렬 음성인식

윤지현°, 홍성태[1], 정상화, 김형순†
부산대학교 컴퓨터공학과, 부산대학교 전자공학과†

Parallel Speech Recognition on Distributed Memory Multiprocessors

Ji-Hyeon Yun, Sung-tae Hong†, Sang-Hwa Chung, Hyung-Soon Kim†
Dept. of Computer Engineering, Pusan National University
Dept. of Electronic Engineering, Pusan National University†

요 약

본 논문에서는 음성과 자연언어의 통합처리를 위한 효과적인 병렬 계산 모델을 제안한다. 음소모델은 continuous HMM에 기반을 둔 문맥종속형 음소를 사용하며, 언어모델은 knowledge-based approach를 사용한다. 또한 계층구조의 지식베이스상에서 다수의 가설을 처리하기 위해 memory-based parsing 기술을 사용하였다. 본 연구의 병렬 음성인식 알고리즘은 분산메모리 MIMD 구조의 다중 Transputer 시스템을 이용하여 구현되었다. 실험을 통하여 음성인식 과정에서 발생하는 speech-specific problem의 해를 제공하고 음성인식 시스템의 병렬화를 통하여 실시간 음성인식의 가능성을 보여준다.

1. 서론

음성에 있어 사용되는 두 가지 지식원은 음소모델과 언어모델이다. 입력된 음성 신호적 특성을 모델링하는 음소모델로 Hidden Markov Model(HMM) 기법 등의 통계적인 방법이 가장 널리 쓰이며, 인식된 어휘인 음절이나 단어간의 언어적 순서 관계를 나타내는 언어모델로는 확률적 언어 모델링과 FSN(finite state network) 언어 모델링 방법등이 사용된다. 음성인식 자체만으로는 인식의 한계를 가지므로 인식된 음성을 언어적 특성이 고려된 형태로 변환하는 작업이 필수적이며, 이러한 음성과 언어처리 기술의 통합처리가 중요한 연구과제가 되고 있다.

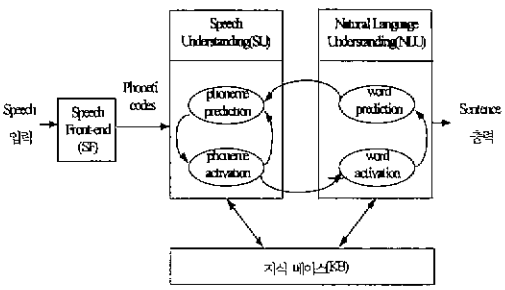
본 논문에서는 음성과 자연언어의 통합처리를 위한 효과적인 병렬 계산 모델을 제안한다. 음소모델은 한국어 음소 변이음(allophone) 기반의 HMM을 사용하였고, 언어모델로써 phoneme, word, syntax 등의 지식원들을 효과적으로 결합할 수 있는 계층구조의 지식베이스를 구축하였다. 광범위한 지식베이스상에서 다수의 가설을 처리하기 위해 memory-based parsing[6] 기술을 채택하여 음성과 자연언어의 통합처리를 위한 병렬 계산 모델을 개발하였다. 또한 음성인식 과정에서 발생하는 insertion, deletion, substitution, word boundary detection등의 speech-specific problem을 분석하고 병렬처리에 의한 해를 제공하였다.

지금까지의 병렬 음성 인식에 대한 연구는 다음과 같다. Giachin과 Rullent[7]가 Transputer-based distributed architecture상에서 병렬 parser를 개발하였고, Chung과 Moldovan[2]은 AI 응용을 위하여 개발된 SNAP MPP 시스템상에 음성과 자연어의 통합처리를 위한 병렬 parser를 개발하였다. 또한 Gijsbert Huijsen[5]은 discrete HMM을 기반으로 nCube2 machine에서 모델링과 음성인식을 시도하였고, AT&T[3]에서는 인식하려는 domain내의 모든 문장을 multi-threading하여 처리하는 음성인식 시스템을 제시하였다.

2. 병렬 음성인식 시스템

2.1. 병렬 음성인식 모델

본 병렬 음성인식 시스템은 <그림 1>에 제시된 바와 같이 speech front-end(SF) 모듈과 natural language understanding(NLU) 모듈, speech understanding(SU) 모듈 및 지식베이스(KB)로 나누어진다. SF 모듈은 화자 독립 연속 음성을 처리하여 phonetic code stream을 SU 모듈에 제공한다. SU 모듈은 SF 모듈에 의하여 공급된 phonetic code를 이용하여 matching phoneme sequence를 찾아낸다. NLU 모듈은 high-level 정보를 이용하여 word candidate를 예측하고 SU 모듈에 의해 인식된 word candidate들을 사용하여 sentence output을 형성한다.



<그림 1> 병렬 음성인식 시스템

2.2. 음소 모델

본 논문에서는 연속 HMM(continuous density HMM) 기반의 문맥독립형(context-independent) 음소 모델인 monophone과 문맥종속형(context-dependent) 음소 모델인 triphone 모델을 사용하여 46개의 변이음을 구성하였다. 그리고 SF 모듈의 인식실험은 Entropic사의 Hidden Markov Model(HMM)기반의 음성인식 Tool인 HTK(HMM Tool Kit) V2.0을 사용하였다.

2.3 언어 모델

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 있으며 문장과 어절을 이루는 단위

본 연구는 1996년도 한국학술진흥재단 공모과제 연구비 지원에 의한 결과임

가 된다. 이와 유사하게 phoneme sequence(PS)는 word를 이루는 단위가 되는 것으로 하나의 concept node에 부착되며 계층구조의 가장 하위 계층을 형성한다.

2.4. 병렬 음성인식 machine

본 연구에서 사용한 병렬 음성인식 machine은 호스트 컴퓨터와 병렬 컴퓨터로 구성된다. 호스트 컴퓨터는 음성인식에 사용되는 입력 데이타를 공급하고 전체 시스템을 관리하는 역할을 담당한다. 병렬 컴퓨터는 호스트로부터 들어오는 입력에 대한 다수의 후보해를 동시에 평가한다. 본 연구에서는 이러한 시스템을 구성하기 위해 분산 메모리 MIMD 구조의 병렬 컴퓨터로써 가격 대 성능비가 우수하고 병렬 프로그램 개발환경이 뛰어난 다중 Transputer 시스템을 사용하였다. 다중 Transputer는 1개의 root 프로세서 (T_R)와 16개의 processing element (Tc)로 구성되어 있다. 여기서 root 프로세서는 host와 Tc들간의 통신을 담당하고, 각 Tc는 실제로 local control에 의해 parsing을 수행하는 역할을 한다.

실험에 사용된 각 프로세서는 Inmos사의 T805 32bit Transputer이다. T805는 다른 프로세서와 통신의 지원하기 위해 4개의 양방향 interprocessor communication link를 가지고 있으며 인접 프로세서와 20Mbits/sec로 background 통신이 가능하다.

3. 병렬 음성인식 알고리즘

3.1. 수행 알고리즘

병렬 음성인식 시스템에서 음소 모듈에서 발생된 다수개의 후보해를 평가하기 위한 수행 알고리즘은 다음과 같다. Root 프로세서는 지식베이스를 적절히 분산하여 각 프로세서의 메모리에 그 일부분을 적재시킨 후, SF 모듈로부터 가설에 대한 N-best candidate를 받아서 시간상으로 정렬시킨다. 이렇게 정렬된 데이터는 병렬 프로세서로 전파되어, 각 프로세서에 적재된 지식베이스상에서 병렬 parsing을 수행한다. 만약 가설을 이루는 CSE들이 여러 프로세서의 메모리에 분산된 경우, 자신이 가지고 있는 CSE에 대한 parsing이 끝난 프로세서는 다음 CSE를 보유한 프로세서에게 그 시점까지의 정보를 넘겨주고 이때부터는 정보를 넘겨받은 프로세서가 그 가설에 대한 parsing을 계속 수행한다. 모든 데이터에 대해 수행을 마치면 root 프로세서는 이 값을 받아 후보해중에서 가장 점수가 높은 것을 선택한다.

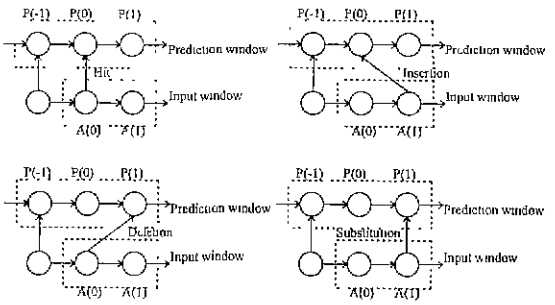
3.2 병렬 parsing 방법

지식베이스상에서 일어나는 memory-based parsing 과정은 parallelism을 최대한 활용하기 위한 추론 메카니즘으로 기본적으로 top-down prediction과 bottom-up activation을 통해 이루어진다. 음소 모듈로부터 phonetic code가 들어오면 각 프로세서내의 지식베이스상에서 상위 계층으로부터 밑으로 prediction이 전파된다. 처음에는 모든 CSR과 그것이 포함하는 하위 계층의 첫번째 CSE와 PSE가 가설로서 prediction되지만 phonetic code의 일치하는 node에만 activation이 일어난다. Activation이 일어난 node는 연결된 link를 통해 다음 node로 prediction을 진파시킨다.

3.3. Window를 이용한 speech-specific problem의 해결

병렬 parsing 과정에서 중요하게 고려되어야 할 부분은 음성인식 과정에서 생기는 insertion, deletion, substitution 등의 speech-specific problem이다. SF 모듈이 제공하는 phoneme sequence는 이러한 문제점을 내포하고 있으므로 align될 수 있는 범위를 확대하여 이러한 문제점을 해결하기 위해 window를 사용한다.

Prediction window는 parsing의 진행 과정에서 activation이 일어날 수 있는 범위를 가리키는 phoneme list로써 현재 prediction된 node와 그 앞뒤로 이어진 node로 구성된다. Input window는 다음에 들어올 phonetic code까지 align의 범위를 확장시키기 위해 사용되며, 이는 두 개의 phonetic code로 이루어진 list이다. <그림 2>는 window를 사용하였을 때 발생가능한 음소열에 대한 alignment를 보여준다. Window내의 P(0)와 A(0)은 현재 prediction된 node와 activation된 node를 가리킨다. 두 window내에서 화살표로 연결되는 node들이 일치할 때 그에 해당하는 alignment로 해석된다. 또한 여기서 alignment 판단시 각 음소에 대한 인식 정확도 및 insertion, deletion, substitution의 발생 빈도를 정보로서 사용하여 좀 더 신뢰성있는 판단을 유도한다.



<그림 2> 발생 가능한 음소열의 alignment

3.4. Scoring 기법

Window와 code/phoneme statistics를 활용하여 지식베이스상의 phoneme과 phonetic code 사이에 alignment가 결정되면, alignment의 정도에 따라 후보해의 score에 penalty가 더해진다. Hit의 경우 정상적인 align이므로 penalty가 없지만, insertion, deletion, substitution의 경우 현재 phonetic code가 가지는 점수에 penalty를 부여한다. 이는 HMM을 통해 들어오는 input stream이 문장의 원래 phoneme sequence와 다를수록 많은 penalty가 부여하여 다수의 후보해중에서 가장 올바른 문장을 얻기 위한 비교자료로 사용한다.

4. 실험

본 연구에서 제시한 병렬 음성인식 알고리즘의 평가를 위해 사용한 음성 데이터베이스는 KOREAN SPEECH DB(ETRI Wonkwang SPEECH DB)[4]를 컴퓨터 비서 에이전트 영역이다. 본 연구에서는 남성화자 56명이 1인당 77문장씩 발음한 데이터를 이용해 훈련했으며, 평가를 위해서는 남성화자 6명이 1인당 77문장씩 발음한 데이터를 대상으로 음소 및 음성에 대한 인식률과 수행시간을 비교하였다. 실험에 사용된 T805 Transputer는 25MHz의 속도로 동작한다.

4.1. 음소 및 문장 인식 결과

본 논문에서는 HMM 모델 구조로 3개의 state를 사용하고, 문법 구조로는 우리말의 음소구성방법을 이용한 grammar를 사용하였다.

<표 1> SF 모듈 인식 결과

Alignment	N	H	I	D	S
개수	35893	30833	8632	831	4229

환경	음소제약을 두지 않은 경우	음소 제약을 둔 경우
State 3개 Triphone	%Corr=84.86%, %Acc=58.74%	%Corr=85.90%, %Acc=61.65%

<표 1>은 전체 음소에 대한 각 alignment 개수와 이를 통해 얻은 변이음 인식 결과이다. 우리말의 음소구성방법을 이용하여 음소제약을 두었을 경우에 더 우수한 인식결과를 나타내었으며, 이들 표에서 인식성능 평가수단인 %Correct와 %Acc(Accuracy)는 다음과 같은 수식으로 주어진다.

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

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

여기서 N은 전체 변이음 개수, D는 deletion된 변이음 개수, S는 substitution된 변이음 개수, 그리고 I는 insertion된 변이음 개수를 의미한다.

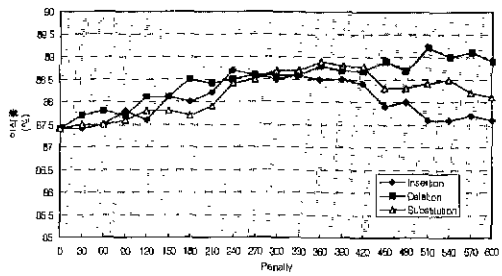
<표 2> 문장 및 단어 인식률

	총 문장	인식된 문장	문장 인식률
지식베이스를 이용하였을 때	462	398	91.6 %
HTK tool을 사용하였을 때	462	360	77.9 %

<표 2>는 본 연구에서 제시하는 병렬 음성인식 시스템에서 얻은 문장 인식률과 HTK tool을 사용한 결과를 비교한 값으로, 실험 4.2에서 얻어진 최적의 penalty 값을 사용하였다. 실험 결과, <표 1>에 나타난 음소 인식률은 85.90%밖에 되지 않지만 이 음소들이 언어처리 모듈을 거치면서 전체 문장에 대한 인식률이 91.6%로 향상된 것을 알 수 있다. 또한 HTK tool만을 사용한 결과보다 13.3% 더 나은 성능을 보였다.

4.2 Penalty 변화에 따른 인식률의 변화

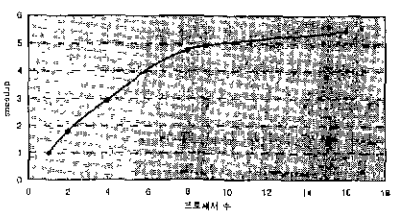
<그림 3>는 3.4절에서 설명한 penalty 값을 결정하기 위한 실험 결과로써, insertion, deletion, substitution에 대한 penalty를 0부터 600까지 각각 변화시키면서 비교한 값이다. 그림에서 보는 바와 같이 인식률을 향상시킬 수 있는 대략적인 penalty는 insertion, deletion, substitution에 따라 각각 240-270(α), 510-540(β), 360-390(γ)으로 결정되었다. 이러한 범위내에서 α, β, γ를 좀 더 정밀하게 변화시켜 가면서 가장 좋은 인식률이 나올 때의 penalty를 얻을 수 있었다. 이 값을 <표 1>의 alignment 개수와 비교해 보았을 때, insertion의 경우와 같이 전체 음소에 대한 발생 빈도가 많을수록 상대적으로 작은 penalty 값을 가진다는 것을 알 수 있다.



<그림 3> penalty 변화에 따른 인식률

4.3. Processor 수의 변화에 따른 수행시간의 향상

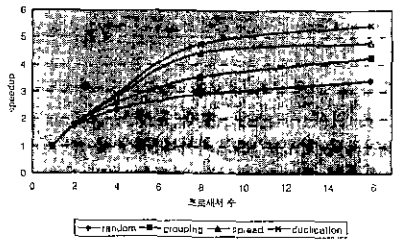
Transputer 상에서 프로세서 수의 증가에 따른 수행속도의 향상은 <그림 4>와 같다. <그림 4>는 프로세서를 1개 사용했을 때의 수행시간을 1로 두었을 때 프로세서 수의 증가에 따른 speedup을 보여주고 있다. 프로세서 1개를 사용하였을 때보다 16개의 프로세서를 사용하였을 때 5.42배의 향상을 보인다. 프로세서 수가 8개 이상으로 늘어나도 수행 속도가 linear하게 향상되지 않는 이유는 지식베이스상의 CSE에서 동시에 평가될 수 word의 수가 평균적으로 6개를 넘지 않기 때문이다. 즉 병렬처리로 인한 parallelism은 동시에 평가될 수 있는 word 개수에 의해 한정된다.



<그림 4> processor 수에 따른 speedup

4.4. 지식베이스의 분산 적재 방법에 따른 수행시간의 향상

<그림 5>는 각 프로세서에 분산 적재되는 지식베이스를 달리함으로써 나타나는 수행시간의 변화를 실험하여 그 결과를 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 group를 모든 node에 duplicate시켰을 때(duplication)를 나타낸다. 그림에서 보는 바와 같이 duplication의 경우가 random하게 적재한 경우보다 프로세서를 16개 사용하였을 때 1.58의 성능향상을 볼 수 있다. 또한 한 node에 같은 word group의 단어를 모두 적재시켰을 때보다 word group의 각 단어를 분리시키는 것이 더 좋은 성능을 보인다. 이를 통해 일정한 시점에서 동시에 평가될 수 있는 word를 인접 프로세서에 분리하여 적재시킴으로써 더 좋은 성능향상을 꾀할 수 있으며 동시에 많이 쓰이는 word를 모든 node에 적재시켜 communication overhead를 줄이는 방향이 더 나은 결과를 보인다는 것을 알 수 있다. 하지만 대용량의 지식베이스의 경우 duplication을 적용할 범위를 결정하는 것이 아직 문제로 남아있다.



<그림 5> 분산 적재 방법에 따른 speedup

5. 결론

본 논문에서는 음성과 자연어의 통합처리를 위한 병렬 음성인식 알고리즘을 제시하였다. 음소모델은 *continuous HMM*을 사용하였고 언어모델은 계층적 지식베이스를 기반으로 하였다. 병렬 음성인식 알고리즘은 Transputer 시스템상에서 구현되었으며 음성인식 과정에서 발생하는 *speech-specific problem*을 분석하였다. 또한 음성인식 시스템의 병렬화를 통하여 실시간 음성인식의 가능성을 보여주었다. 앞으로의 연구 방향은 대용량 음성 DB를 사용하기 위한 knowledge representation 기술의 개발과 SP 모듈과 SU, NLU 모듈의 밀결합에 의해 성능을 향상시키는 방안에 대해 지속적인 노력이 필요하다.

참 고 문 헌

[1] J. D. Markel, A. H. Gary, Jr, Linear Prediction of Speech, Springer-Verlag, 1776.

[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] Steven Phillips, Anne Rogers, "Parallel Speech Recognition", *In Proceedings of EUROSPEECH-97*, pp.135-138, 1997

[4] 이용주, 음성 데이타베이스 설계 및 제작, 용역결과보고서, 한국전자통신연구소, 1996년, 8월.

[5] G. Huijsen, "Parallel Implementation of Hidden Markov Models on the nCUBE", M.Sc. thesis, Alparon report, nr. 96-03, Delft University of Technology, 1996

[6] C. Stanfill and D. Waltz, "Toward Memory-Based Reasoning", *Communications of the ACM*, vol.29, no.12, 1986.

[7] E. P. Giachin and C. Rullent, "A Parallel Parser for Spoken Natural Language", Proceedings of IJCAI, pp.1537-1542, 1989.