2000년도 한국음향학회 학술발표대회 논문집 제19권 제1(s)호

1500 단어 실시간 화자 독립 음성인식 시스템

이 강성
광운대학교 컴퓨터 공학과

Realtime Speaker Independent Speech Recognition System of 1500 Words

Gang Seong Lee
Computer Engineering Dept., Kowangwoon Univ.
gslee@mail.gwu.ac.kr

요 약

본 논문은 중규모 어휘인 1500여 단어 실시간 화자 독립 단독어 음성인식 시스템에 대해서 기술한다. 음향 모델은 HMM을 이용하였으며, 음소 모델은 문맥 종속 모델인 트라이폰을 사용하였다. 이 시스템은 텍스트로부터 쉽게 사전을 구성할 수 있는 유연성을 갖는다. 선정된 단어는 주식시장에 상장되어 있는 1456개의 회사명으로 비교적 혼동하기 쉬운 단어들을 많이 포함한 사전이다. 실시간 처리를 위한 알고리즘들 중 인식율을 크게 저하시킬 가능성이 있는 기법들은 제외하였다. 여기에 트리 빔과 음소 빔을 적용하면서 topN을 적용하였으며 새로운 스코아 캐쉬 기법을 고안하였다. 특별히 스코아 캐쉬 기법은 인식율에는 전혀 영향을 미치지 않으면서 계산량을 38%나 줄여주었다. 이런 기법들을 적용하여 실시간 음성인식을 구현할 수 있었다. Intel 450M CPU가 장착되어 있는 리눅스 시스템에서 평균 1.98초의 응답 시간을 보였다.

1. 서 론

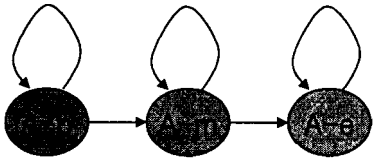
HMM은 음성인식 기법 중에서 가장 보편적으로 사용되는 기법이다. 이 기법은 비교적 많은 메모리와 계산 시간을 필요로 하여 적절한 비용의 시스템 구축에 어려운 점이 없지 않았으나, 시간이 지남에 따라 하드웨어의 가격 하락과 CPU 계산 성능의 향상, 메모리 칩의 대용량화에 힘입어 대중화된 PC에서도 무리없이 처리가 가능한 수준까지 도달하였다.

이에 본 연구실에서는 반연속(semicontinuous) HMM을 이용한 새로운 음성인식 시스템을 설계 중에 있으며, 그 첫 번째 중간 결과로 1500단어의 화자 독립 단독어 음성인식 시스템에 대한 개괄적인 시스템 구조를 설명한다.

시스템은 학습 모듈과 인식 모듈로 나누어지며, 학습 모듈은 음성 특징 파라미터 계산부, 문맥 독립(context independent) 음소 모델 작성부, 음소 레이블링부, 문맥 집단화(context clustering)[2]을 이용한 문맥 종속(context dependent) 음소 모델 작성부등이 있다.

인식 모듈은 음성 특징 파라미터 추출부, 검색(search) 알고리즘 부가 있다. 음소 파라미터를 위한 코드북(codebook)과 분포 무게 값(distribution weight)을 저장하는 자료 구조가 있다. 각 코드워드는 평균 벡터와 분산 값을 갖는 가우시안 확률 파라미터이다. 분포 무게 값은 각각의 가우시안 확률 파라미터에 대한 무게 값이다. 하나의 음소로부터 파생된 여러 개의 폴리폰(polyphone)은 많은 경우 하나의 코드 북을 공유한다. 하지만 별개의 분포 무게 값을 가지므로 다른 확률 값을 표현한다.

각 음소는 세 개의 상태로 표현되며(예: a-b, a-m, a-e), 각 상태의 천이 위상 구조는 단순한 Left-to-right 모델이다.



본 논문에서는 이러한 구조를 바탕으로 하여 연산 시간을 줄이기 위해 시스템에 적용된 기법들을 설명한다. 시간을 줄이기 위한 여러 가지의 기법들 중에서 본 시스템에서는 프레임 건너띄기(frame skipping)과 같은 인식율을 현저하게 저하시킬 가능성이 있는 기법들은 우선 제외하였다.

적용된 기법들은 일반적으로 잘 알려진 topN 기법을 적용한 빔 검색 기법이다. 또한 계산된 확률값(score)를 저장한 후 재사용하는 캐쉬 기법을 사용함으로 계산시간을 크게 줄였다.

2. 적용된 계산 시간 감축 알고리즘

1) 트리 구조의 단어 사전

전체 어휘에 대한 선형 검색 시간을 줄이기 위한 보편적인 음소 구성 방법이 트리 구조이다. 트리 구조는 동일한 음소로 시작하는 단어들을 모아서 트리 형식으로 구성한 것이다. 다음 그림은 문맥 독립 음소들을 이용하여 트리를 구성한 예이다.[3]

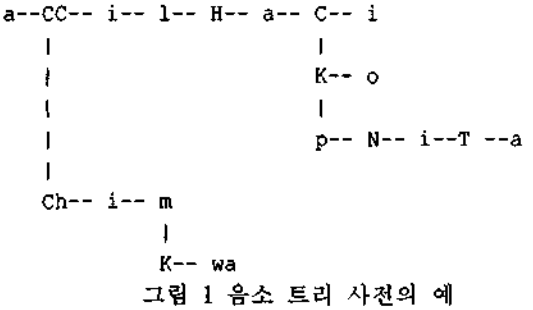


그림 1 음소 트리 사전의 예

2) 빔 검색

본 논문에서는 트리 검색을 위해서 두 가지의 빔을 사용했다. 다양한 빔은 상황에 따라서 다른 임계값을 적용하는 것이 유용한가를 보기 위함이다.

가) 한 트리 내에서의 가지치기(pruning)

하나의 루트를 갖는 노드들 중에서 가장 높은 가능성을 갖는 노드 값을 기준으로 일정 값을 넘는 모든 가지를 비활성화 시킨다.[3]

나) 트리와 트리 간에 적용:

모든 트리 중에서 가장 높은 가능성 값을 기준으로, 임의의 트리의 최고 가능성이 이 보다 일정 값을 넘는다면(가능성이 낮아진다면) 트리 전체를 비활성화 시킨다.

위 나)항을 적용한 이유는 다른 트리의 임계값으로 어떤 트리의 노드를 비활성화 시켰을 때 가능성은 현재 시점으로 좀 떨어지지만 이후에 높은 가능성을 가질 수 있는 노드를 살려둔다면 유용하지 않을까 하는 관점에서이다.

3) topN 기법 적용

빔을 적용해도 활성화되는 노드들이 시간이 감에 따라서 크게 늘어난다. 따라서 이들을 적절하게 제어할 수 있는 방법이 필요하다. 활성화된 노드들의 최대 수를 일정 수 이하로 유지시킴으로 검색 속도를 증가시킬 수 있다. 한 예로 topN=100으로 이 기법을 적용했을 때 약 33.0%의 계산시간이 감축되었다.

4) 제안된 스코아 캐쉬 기법

시간 동기화 비터비(Viterbi) 검색 방법에서 한 프레임이 입력되면 그에 따라 전체 노드에 대해서 계산을 수행한다. 이 때 다른 노드들 간에 같은 음소 모델을 공유하는 경우가 많이 발생한다. 그림 1에 보면 두 개의 단어 '아이' 와 '이아' 는 매번 프레임이 입력될 때마다 계산되므로 약 두 배의 시간을 소비하게 된다. 하나의 상태를 계산할 때 그에 대한 확률값을 저장하고 있다가, 다음에 요구할 때 재사용 하도록 캐쉬 영역에 저장한다면 계산 량을 크게 줄 일 수 있다. 더욱이 하나의 음소는 여러 개의 상태로 나누어지지만 이들은 하나의 코드북을 공유하고 있다. 따라서 분포 확률만 누적하면 되므로 계산 효과는 상당히 클 수 있다.

스코아 캐쉬 기법은 두 가지 수준에서 구현이 가능하다. 첫 번째는 각 음소의 부 모델별 캐쉬를 사용할 수 있고 두 번째는 각 음소의 부 모델별 캐쉬 및 코드북에 대한 캐쉬를 사용할 수 있다. 단, 첫 번째 방법에서는 각 입력 프레임에 대한 확률값 하나만 캐쉬에 저장하면 되므로 메모리에 큰 부담이 없으나 두 번째 방법에서 코드북에 대한 캐쉬는 N개의 가우시안 파라미터에 대한 각각의 확률 값을 저장하고 있어야 하므로 메모리 부담이 있다.

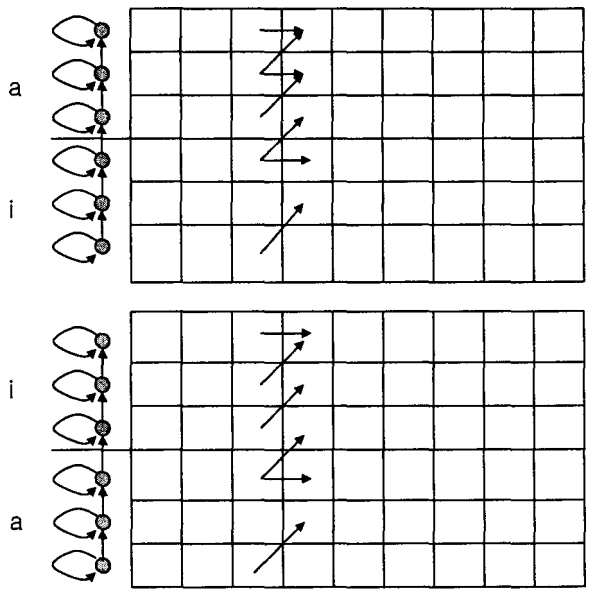


그림 2 시간 동기 비터비 검색시 확률값 계산

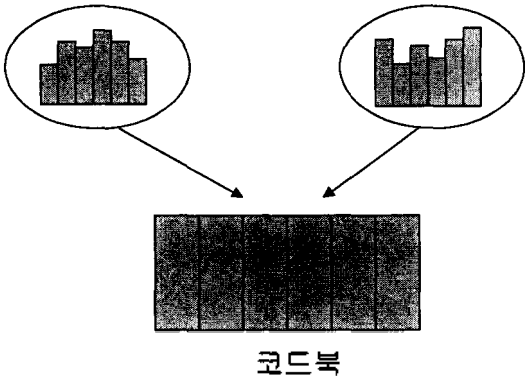


그림 3 두 개의 상태가 하나의 코드북을 공유

구현에 있어서도 캐쉬의 값이 코드북의 id 나 분포 확률 값의 id 에 따라서 언제나 정렬된 상태로 있어야 하므로, 선형 리스트 보다는 인덱스(indexed) 트리나 해쉬 기법을 적용하는 것이 빠른 구현에 도움이 될 것이다.

본 연구에서는 해쉬 기법을 이용하여 우선 첫 단계 수준의 기법만 적용하였다. 즉, 부 음소 단위 (상태 단위)의 캐쉬를 적용하였다.

3. 실험 및 고찰

1) 데이터베이스

음소 모델을 학습시키기 위해서 사용한 음성은 뉴스 음성이다. 데이터베이스 KBN01은 총 16회분의 뉴스로 구성되어 있으며 그 중에서 앵커와 기자들만의 목소리로 트라이폰 음소 모델을 만들었다. 앵커와 기자들의 수는 약 180여명이다.

2) 신호처리

음성 샘플은 비디오 테입으로부터 44.1KHz의 MPEG 포맷으로 변환된 후 다시 16KHz PCM 파일로 다운 샘플링되었다. 각 샘플은 16비트이다.

특징 벡터를 계산하기 위해서 프레임 윈도우는 20ms, 윈도우의 이동은 10ms씩 되었다. 다음의 43차원의 특징 벡터가 계산되었다.

mel ceptrum 13 차
delta melcepstrum
delta delta melcepstrum
zero crossing rate
logpower
delta logpower
delta delta logpower

그리고 나서 K-L 확장 법을 이용하여 24차원으로 줄였다[4].

3) 음향 모델링

모든 문맥 종속 음소모델은 묵음을 제외하고는 left-to-right 위상을 갖고 있다. 묵음은 하나의 상태만을 갖는다. 2000개의 부 음소 모델이 만들어 졌다. 코드북은 16개의 가우시안 파라미터를 갖으며 대각원 공분산 (diagonal covariance)값이 사용되었다.

4) 1456개의 단어 목록

1456개의 단어 목록은 주식 정보를 제공하는 웹 상에서 상장된 회사명을 취한 것이다. 유사한 단어도 다수 포함되어 있다. 단어의 평균 음절수는 4.32이다.

5) 인식율

학습에 참여하지 않은 남성 화자 5명이 각각 600개씩의 단어를 발성하여 실험하였다.

두 가지의 beam 값과 topN 값을 각각 200, 200, 100으로 설정하였을 때의 인식 결과는 평균 94.5%가 나왔다.

6) 스코아 캐쉬 기법

앞서 설명한 첫 단계의 스코아 캐쉬 기법을 적용했을 때 계산량 감축에 대한 간단한 실험을 했다.

캐쉬를 적용하지 않았을 때의 평균 인식 시간은 3.23초로 캐쉬를 적용했을 때의 1.99초는 약 38.4%의

시간절약이 된 것이다. xRealTime 단위는 다음과 같이 정의되는데,

xRealTime = 인식 시간 / 입력 음성의 길이

음성신호의 길이에 대한 인식 시간의 비를 나타낸다. xRealTime은 캐쉬를 적용하지 않았을 때 3.13, 적용했을 때 1.92로 38.7%의 비슷한 비율의 시간절약이 된 것을 알 수 있다.

5) topN을 적용했을 때의 시간 절약

topN을 적용했을 때와 적용하지 않았을 때의 시간은, 적용했을 때 3.00초, 적용하지 않았을 때 1.99초로 약 33.6%의 시간이 절약되었다.

4. 결론

본 논문에서는 1456개 단어의 화자 독립 단독어 음성인식 시스템을 개괄적으로 기술했으며, 실시간 인식을 위해서 적용된 여러 기법들을 설명했다. 94.5%의 인식율을 보였다. 특별히, 시간을 줄이기 위한 스코아 캐쉬 기법을 소개했으며 간단한 실험을 통하여 약 38%의 계산 시간이 줄어든 결과를 보임으로 그 효용성을 입증하였다. 계산 시간은 현재 1.99초이었다. 사용된 시스템은 Intel Pentium III 450MHz CPU가 장착된 Linux 시스템이며, CPU를 조금 더 빠른 것을 사용한다면 얼마든지 더 빠른 응답을 낼 수 있다.

참고 문헌

[1] Lawrence Rabiner, Biing-Hwang Juang, Fundamentals of Speech Recognition, Prentice Hall, 1993

[2] Ivica Rogina, "Automatic Architecture Design by Likelihood-Based Context Clustering With CrossValidation", Proceedings of Eurospeech-97, 1997

[3] H. Ney, et al. "Improvments in beam search for 10000-word continuous speech recognition", Proc. ICASSP'92, pp.13-16

[4] Keinosuke Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press Inc. 1990