한국음향학회지 제20권 제6호 pp. 46~50 (2001)

시간 동기 비터비 빔 탐색을 위한 인식 시간 감축법

Recognition Time Reduction Technique for the Time-synchronous Viterbi Beam Search

이 강 성*
(Gang Seong Lee*)

*광운대학교 컴퓨터공학과
(접수일자: 2001년 5월 2일; 채택일자: 2001년 6월 25일)

본 논문은 HMM (Hidden Markov Model) 음성 인식 시스템에 적용할 수 있는 새로운 인식 시간 알고리즘인 스코아 캐쉬기법을 제안한다. 다른 많은 기법들이 인식 시간을 줄이면서 계산량을 줄이기 위하여 어느 정도의 인식율 저하를 감수하는 반면에 제안하는 스코아 캐쉬기법은 인식율 저하를 전혀 일으키지 않으면서 인식 시간을 상당량 줄일 수 있는 기법이다. 단독어 인식 시스템에 적용 가능할 뿐 아니라 연속어 인식에도 적용이 가능하며, 기존에 이미 설계된 인식 시스템의 구조를 전혀 훝트리지 않고 간단히 하나의 함수만 대치함으로서 인식 시간을 크게 감축할 수 있다. 또한 기존의 계산량 감축 알고리즘과 함께 적용 가능하므로 추가의 계산량 감소를 얻을 수 있다. 스코아 캐쉬 기법을 적용한 결과 최대 54% 만큼 계산량을 줄일 수 있었다.
핵심용어: 음성인식, HMM, 인식 시간 감축, 검색, 스코아 캐쉬, 연속어 인식
투고분야: 음성처리 분야 (2.5)

This paper proposes a new recognition time reduction algorithm, Score-Cache technique, which is applicable to the HMM-base speech recognition system. Score-Cache is a very unique technique that has no other performance degradation and still reduces a lot of search time. Other search reduction techniques have trade-offs with the recognition rate. This technique can be applied to the continuous speech recognition system as well as the isolated word speech recognition system. We can get high degree of recognition time reduction by only replacing the score calculating function, not changing any architecture of the system. This technique also can be used with other recognition time reduction algorithms which give more time reduction. We could get 54% of time reduction at best.
Keywords: *Speech recognition, HMM, Recognition time reduction, Search, Score cache, Continuous speech recognition*
ASK subject classification: *Speech signal processing (2.5)*

I. 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 (Central Processing Unit) 계산 성능의 향상, 메모리 칩의 대용량화에 힘입어 대중화된 PC에서도 무리없이 처리가 가능한 수준까지 도달하였다.

하지만 여전히 많은 어휘의 단어를 인식하기 위해서는 많은 계산량을 필요로 하여, 계산량을 줄이기 위한 연구가 꾸준히 이루어져 왔다.

언어 모델을 관련시키지 않고 계산량을 줄이기 위한 기법들로는 다음과 같은 것들이 있다.

책임저자: 이강성 (gslee@mail.gwu.ac.kr)
139-701 서울시 노원구 월계동 447-1
광운대학교 컴퓨터공학과
(전화: 02-940-5284; 팩스: 02-914-4751)

1. 빔 탐색 (Beam Search)

잘 알려진 기법으로 현재의 시간 프레임에서 확률이 가장 높은 상태를 기준으로, 이 확률보다도 일정한 값 이하로 떨어지는 다른 상태 (states)들을 검색 대상 경로에서 제외시키는 것이다.

2. 트리 탐색[1]

이 또한 대부분의 시스템에 적용되고 있는 기법으로, 어휘 사전을 선형적이 아닌 공통 음소를 공유하는 트리로 구성하여 탐색 공간을 크게 줄인다.

3. 음소의 룩어헤드 (look-ahead) 기법[1]

한 음소의 마지막 상태에서 다른 음소로 전이가 가능한지의 여부를 미리 판단하기 위하여, 앞으로 계산될 몇 개의 프레임으로 그 가능성을 미리 계산하여, 기준 값에 미달하는 음소로의 전이를 금지하는 방법이다.

4. 전후방 (forward-backward) 탐색 알고리즘[2]

빔 탐색과 함께 적용되는 기법으로, 검색을 전방향 (forward)으로 진행함과 동시에 후방향 (backward)으로도 함께 진행하여 전체적인 탐색 공간을 크게 줄인다.

5. BBI (Bucket Box Intersection) 알고리즘[3]

각 코드북 벡터의 가우시안 확률 분포도에서 일정 수준 이하의 확률을 갖는 영역을 제외하고 남는 공간의 사각 영역을 Bucket Box Intersection이라고 하는데, 어떤 벡터가 입력되었을 때 이 벡터를 포함하는 사각 영역 코드북 벡터에 대해서만 계산을 할 수 있도록 트리 형식으로 공간을 분할하는 것이다. 이 기법을 적용했을 때 약 20%의 계산량을 줄일 수 있었다고 한다.

6. 프레임 건너뛰기 (Frame Skipping)[3]

입력 프레임을 일정 비율로 건너뛰면서 인식하는 기법이다.

일부 기법들은 계산량을 줄이기 위하여 약간의 인식율 감소를 감수하기까지 하였지만, 본 논문에서는 인식율의 감소가 전혀 없으면서도 인식 시간을 크게 줄일 수 있는 스코아 캐쉬 기법을 소개한다.

스코아 캐쉬 기법은 시간 동기 빔 검색 (time synchronous beam search)에서 많은 확률 값들이 중복 계산되는 것을 방지하는 알고리즘이다.

II. 스코아 캐쉬 알고리즘

대부분의 음성 인식에 사용되는 음향 모델은 음소를 기반으로 한다. 문맥에 관계없이 음소 심볼 기호마다 각각의 모델을 만드는 문맥 독립형 (context-independen) 모델이 있는가 하면, 어떤 음소의 좌우에 나타나는 음소에 따라 독립적인 모델을 만드는 문맥 종속형 (context-dependent) 모델이 있다 (polyphone 이라고도 한다). 좌우에 한 개씩의 음소를 고려하는 경우를 트라이폰 (triphone), 두 개씩의 음소를 고려하는 경우를 퀸트폰 (quintphone) 이라고 한다. 이들은 보통 3-5개의 상태를 갖는 HMM으로 표현된다.

어휘 사전에 있는 수많은 단어들은 이러한 단위들을 기반으로 구성된다. 즉, 아무리 단어가 많아도, 이러한 모델들이 반복해서 출현하게 되며, 입력 프레임에 대하여 어느 정도 중복 계산이 된다. 이러한 중복계산을 피하기 위하여 어휘사전의 트리 구성이 일반화되어 있으며, 이렇게 트리로 구성할 경우 10000단어의 실험에서 선형 탐색 기법에 비해서 약 1/7 정도로 계산량이 감축되었다고 이미 보고되어 있다[1].

이러한 계산량의 감소는 많은 단어들이 같은 음소로 시작하며 중복 계산되는 음소를 하나로 묶어서 트리형식으로 표현함으로 가능하다. 그러나 이러한 방식은 같은 수준의 음소 모델이 중복되었을 경우에만 계산량 감축이

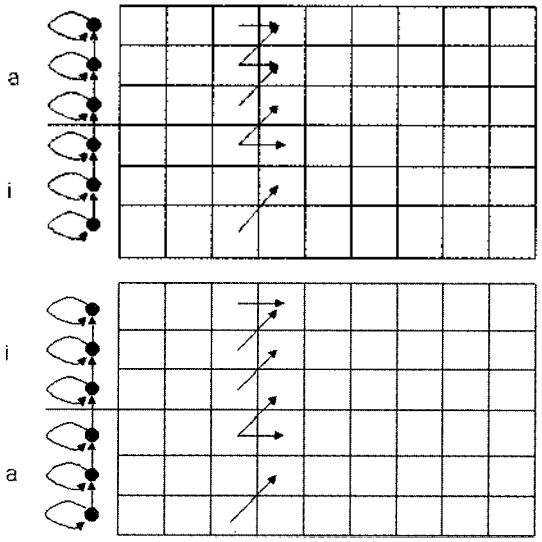

그림 1. 시간 동기 비터비 탐색을 위한 한 시점에서의 연산 (횡축은 프레임을 나타냄)

Fig. 1. An example of search path at a frame for time synchronous Viterbi search (horizontal axis is for frames).

일어난다. 즉, [a b c], [a c b]와 같은 음소 열을 갖는 두 개의 단어인 경우, 처음에 시작 위치가 같은 음소 a만이 트리 안에 공통 부분으로 표현되며, 레벨 (level)이 다른 음소 b와 c는 공유되지 못한다.

그러나 실제로 시간 동기 빔 탐색법이 적용될 때에 이들 다른 레벨의 음소들이 같은 수준에서 비교된다. 즉, 다음의 예를 보자. 단어 w1=[a i]와 w2=[i, a]에서 w1과 w2의 a나 i는 모두 서로 다른 레벨의 음소이지만 현재 시간 프레임에서 모두 계산된다.

따라서, 음소들이 트리의 노드로 공유되지 않더라도, 탐색 시에 중복된 확률 계산은 막을 수가 있다. 이를 위해서 캐쉬 기법을 적용하면 된다. 즉, 어떤 입력 프레임에 대하여 w1의 a에 대한 확률이 계산되었을 때, 그 값이 동일하게 다른 음소 레벨에 있는 동일한 음소인 w2의 a에도 적용된다. 따라서 확률 값 계산은 우선 캐쉬 메모리를 검사하여 계산하고자 하는 확률값이 존재하는가를 검사하고, 없다면 계산해서 캐쉬에 저장한다. 이러한 캐쉬는 새로운 프레임이 입력될 때마다 클리어되어야 한다.

추가로 확장한다면, 음소 룩어헤드 기법을 적용하여 몇 개의 시간 프레임에 걸쳐서 이러한 캐쉬 내용을 저장하고 있을 수 있을 것으로 보인다. 의사코드 (pseudo-code)로 작성된 알고리즘을 다음에 보인다.

```
clear cache
for each frame
.   for each tree
.   for each polyphone
.   .   for each state
.   .   .   search from the cache
.   .   .   calculate score, if not found,
.   .   .   and store in a cache
            ...
.   clear cache
```

2.1. 스코아 캐쉬 기법에 필요한 메모리 양

스코아 캐쉬 기법에 필요한 메모리 양은 음소 모델의 수에 의존한다. 전체 음소 모델의 수를 M이라 하고, 각 음소가 N 개의 상태를 갖는다고 한다면 최대의 캐쉬 필요량은 $M \times N$ 이다. 예를 들어 M=2000, N=3 인 경우에 6000개의 실수 값을 저장할 공간이 필요하며, 각 경우에 모델 식별을 위한 id 공간 (최적화되었을 경우 2바이트로 가정하자)을 생각한다면, 6000 바이트 * (8 (double의 바이트 수) + 2 (id)) = 60,000 바이트이므로 최대 60K 정도로 보면 무리가 없다. 그리고 이것은 최대 필요량에 불과하며, 빔 기법이 많은 수의 트리나, 가지를 탐색 대상에서 제외하므로 실제로는 이것보다 훨씬 적은 메모리 량이 요구된다. 또한, 이 크기는 단지 음소 모델의 수에만 의존하며, 사전의 크기와는 전혀 무관한 수치이므로, 대어휘 인식에도 적용되어도 메모리는 전혀 문제가 되지 않는다.

2.2. 인식 시간의 감축

인식 시간의 감축이 어느 정도 일어날 것인가 하는 문제는 어휘의 수, 단어들의 유사도, 단어의 길이에 의존한다. 단어의 길이는 어휘가 단어 단위로 등록되어 있을 경우 평균 길이 값은 거의 일정하다고 가정할 수 있으므로 무시해도 좋은 인자이다. 인식 시간 감축은 어휘의 수와 단어들의 유사도에 의존한다고 할 수 있다. 어휘의 수는 많을수록 감축률이 높고, 단어 사이의 유사도도 높을수록 감축률이 높지만, 단어의 유사도는 어휘의 크기가 어느 정도 크게 되면 별 영향을 끼치지 못하는 인자이므로 결국 인식 시간의 감축은 어휘의 수에만 의존한다고 할 수 있다.

III. 실험 및 고찰

3.1. 데이터 베이스

16개의 KBS 9시 뉴스를 녹음해서 (총 13.7시간) 16 kHz 샘플링 주파수로 16비트 PCM 파일로 변환하였다. 음성 자료는 몇 개의 문장을 갖는 세그먼트로 구분되었으며 화자 이름과 화자 그룹, 잡음 레벨 정보가 각 세그먼트에 정보로 추가되어 데이터베이스에 저장되었다.

화자는 앵커, 취재기자, 인터뷰한 사람들로 구분된다. 앵커는 4명 (남자 2명, 여자 2명)이고, 취재기자는 총 191명 (남 183명, 여 8명) 이고 인터뷰한 건수는 774명 (남 633명, 여 141명)이었다. 전체 인원수는 969명으로 남자 818명, 여자 151명이었다.

3.2. 음향 모델

16개의 뉴스 중에서 14개의 뉴스가 학습을 위해 사용되었고 두 개의 뉴스는 시험을 위해서 사용되었다. 음향모델은 앵커스피커 전체와 취재 기자 전체에 대해서 만들어졌다. 각 세그먼트의 잡음 레벨은 그 정도에 따라서 N0-N4까지로 나누어졌는데 음향 모델 학습에 포함시킨 세그먼트는 N0-N3였다. N4는 알아듣기 힘들 정도의 심한 잡음이 배경에 깔려있는 경우이다.

만들어진 인식 시스템의 전체 트라이폰의 수는 2000개이다. 하나의 트라이폰은 세 개의 상태로 구성되며 기본적인 좌-우 (left-to-right) 모델을 사용한다[5]. 하나의 상태는 16개의 코드 벡터로 구성된다[6].

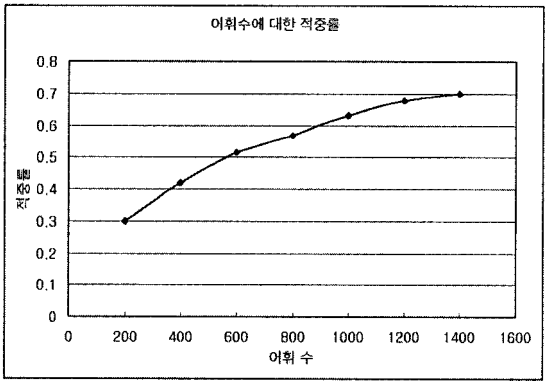

그림 2. 적중률
Fig. 2. Hit ratio.

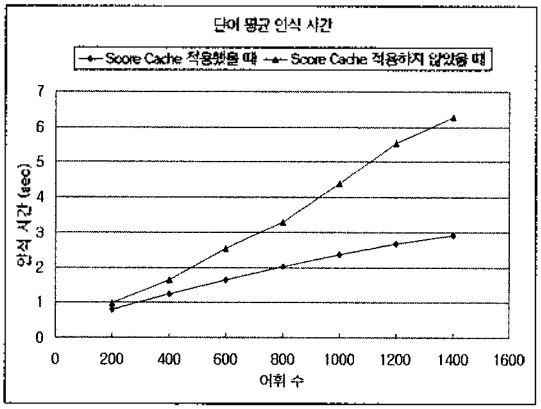

그림 3. 스코아 캐쉬 기법을 적용했을 때와 하지 않았을 때의 단어 인식 시간 비교 (Pentium III 450MHz, Linux)
Fig. 3. Recognition time comparison between score cache technique is applied and not (Pentium III 450MHz, Linux).

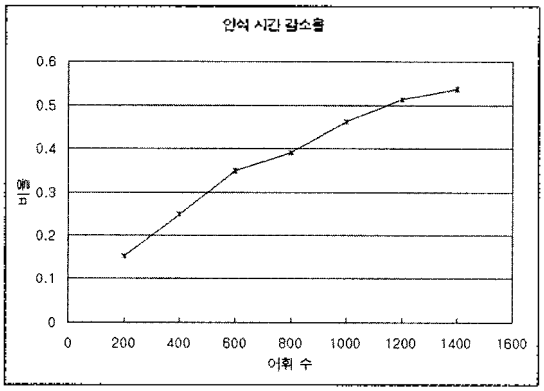

그림 4. 스코아 캐쉬를 적용하였을 때 인식 시간 감소율
Fig. 4. Recognition time reduction ratio when score cache technique is applied.

3.3. 적중률

빔 탐색을 적용한 경우 스코아 캐쉬 기법이 적용되었을 때 적중률 (한번 계산된 스코아 값이 다시 사용되는 비율)을 실험하였다. (빔 탐색을 적용하지 않은 경우는 효용성의 문제와 계산시간이 너무 걸려 실험을 중단했다.) 어휘는 트리 구조로 구성되어 있다. 어휘의 수를 200부터 200씩 증가하면서 100개의 음성 인식에 대한 적중률을 실험하였다. 실험 결과는 그림 2와 같다. 어휘수가 증가함에 따라서 적중률은 0.3에서 0.7까지 높아졌다. 어휘수가 더 증가하면 더 적중률은 높아질 것으로 예상된다.

3.4. 인식 시간 감축률

이론적으로 적중률이 시간 감축량에 비례하기는 하지만, 실제로 인식을 하는데는 오버헤드가 따르기 마련이다. 이 오버헤드는 주어진 부모델 (상태)을 확률 계산 전에 먼저 찾아야 한다는 것과, 찾지 못했을 때 확률값을 계산하여 캐쉬에 저장해야 하는 부담이 있다. 캐쉬에서 계산된 스코아 (확률) 값을 찾기 위해서 해쉬 (hash)기법을 적용하였다. 실제로 구현되었을 때 얼마만큼의 계산 감축이 일어나는가를 (혹은 오버헤드가 얼마나 되는지를) 보기 위하여 실제 인식 실험을 하였다. 인식은 트리 빔 탐색이 적용된 조건하에서 비교되었다. 다음 그림 3에 스코아캐쉬 기법을 적용했을 때의 인식 시간과 적용하지 않았을 때의 인식 시간 비교에 대한 결과를 보인다.

그림 3에서 알 수 있는 바와 같이, 단어 수가 증가할수록 인식 시간의 감축은 증가한다. 이것은 어휘 수가 증가함에 따라 스코아 캐쉬의 적중률이 증가함에 따른 것이다.

그림 4에 보면 인식 시간의 감축률을 알 수 있다. 단어 수가 200인 경우 약 15%의 인식 시간 실제 감축이 일어났지만 (스코아 캐쉬 적중률은 30%였다), 단어 수가 1400인 경우 약 54%의 시간 감축률을 얻었다. 단어수가 1400인 경우의 스코아 캐쉬 적중률은 70%이다.

IV. 결 론

본 연구에서는 음소 모델의 HMM을 이용한 단어 인식 시스템 혹은 연속어 인식시스템에 적용가능한 계산 시간 감축 알고리즘인 스코아 캐쉬 기법을 제안하고 그 성능을 살펴보았다. 이 기법을 적용하면 어휘 수에 따라 차이가 있지만 1400 단어인 경우 적중률이 0.7이상이 되어 계산 시간을 54% 이상 감축시킬 수 있었다.

특히 이 기법은 인식율에 전혀 영향을 주지 않으면서

계산량만을 줄이므로 유용하게 활용이 가능하다. 어휘 사전의 크기가 커지면 커질수록 적중률이 증가하여 대용량의 음성인식 시스템에도 잘 활용될 수 있다.

감사의 글

이 논문은 1999년도 광운대학교 교내 학술연구비 지원에 의해 연구되었음.

참 고 문 헌

1. H. Ney, et al. "Improvments in beam search for 10000-word continuous speech recognition", *Proc. ICASSP' 92*, pp. 13-16.
2. Steve Austin, et al. "The Forward-Backward Search Algorith", *Proceedings of ICASSP' 91*, pp. 697-700.
3. Monika Woszczyna, "Fast Speaker Independent Large Vocabulary Continuous Speech Recognition", CMU, Doctorial Thesis, 1999.
4. 이강성, "1500 단어 실시간 화자 독립 음성인식 시스템", 한국음향학회 하계학술발표대회 논문집, 제 19권 1(s)호, pp. 15-18, July 2000.
5. Keinosuke Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press Inc. 1990.
6. Ivica Rogina, "Automatic Architecture Design by Likelihood-Based Context Clustering With Cross Validation", *Proceedings of Eurospeech-97*, 1997.

저자 약력

● **이 강 성 (Gang Seong Lee)**
한국음향학회지 제20권 제2E호 참조