

# 트리 구조 어휘 사전을 이용한 연결 숫자음 인식 시스템의 구현\*

윤영선(한남대), 채의근(공주대)

## <차 례>

- |                               |                 |
|-------------------------------|-----------------|
| 1. 서 론                        | 3. 구현 및 실험 결과   |
| 2. 연결 숫자음 인식 시스템의 구성          | 3.1. 구현 시 고려할 점 |
| 2.1. 어휘 사전                    | 3.2. 비교 평가      |
| 2.2. 토큰 전달 알고리즘               | 4. 요약 및 결론      |
| 2.3. 어휘 트리와의 유한 상태<br>네트워크 결합 |                 |

## <Abstract>

### Implementation of Connected-Digit Recognition System Using Tree Structured Lexicon Model

Young-Sun Yun, Yi-Geun Chae

In this paper, we consider the implementation of connected digit recognition system using tree structured lexicon model. To implement efficiently the fixed or variable length digit recognition system, finite state network (FSN) is required. We merge the word network algorithm that implements the FSN with lexical tree search algorithm that is used for general speech recognition system for fast search and large vocabulary systems. To find the efficient modeling of digit recognition system, we investigate some performance changes when the lexical tree search is applied.

\* Keywords : Korean digit recognition, lexical tree search

\* 본 연구는 한국과학재단 지역대학우수과학자지원사업 (R05-2003-000-11398-0)과 2003년도 한남대학교 학술연구 조성비 지원으로 수행되었음

## 1. 서 론

일반적인 연속 음성 인식은 음소 모델(phoneme model)이나 단어 모델(word model)과 같은 인식 단위(recognition unit)를 모델링한 후, 탐색 단계에서 언어 모델을 이용하여 인식 단위들의 연결 정보를 파악한 후 최적의 단어 열을 구한다. 그러나, 숫자음 인식은 탐색 단계에서 탐색 공간(search space)을 줄일 수 있는 언어 정보를 이용하기 어렵다는 단점이 있다. 숫자음 인식은 그 사용처에 따라 단어수가 고정되는 경우가 많기 때문에 단어 네트워크를 이용하여 탐색 공간을 줄이는 방법을 사용한다.

대용량 음성 인식 시스템의 경우, 인식하고자 하는 단어 수가 많아지면서 탐색 공간이 확장되어 인식 시간의 증가를 가져오게 된다. 이러한 문제점을 해결하기 위하여 어휘 사전을 모델링할 때 단어를 구성하는 subword를 공유하는 방법이 연구되어 사용되고 있다. 이러한 방식은 기존의 선형 어휘 사전(linear lexicon, flat structured lexicon)에 비교하여 트리 구조 어휘 사전(tree structured lexicon) 또는 접두어 공유 어휘 사전(prefix sharing lexicon)이라고 한다.

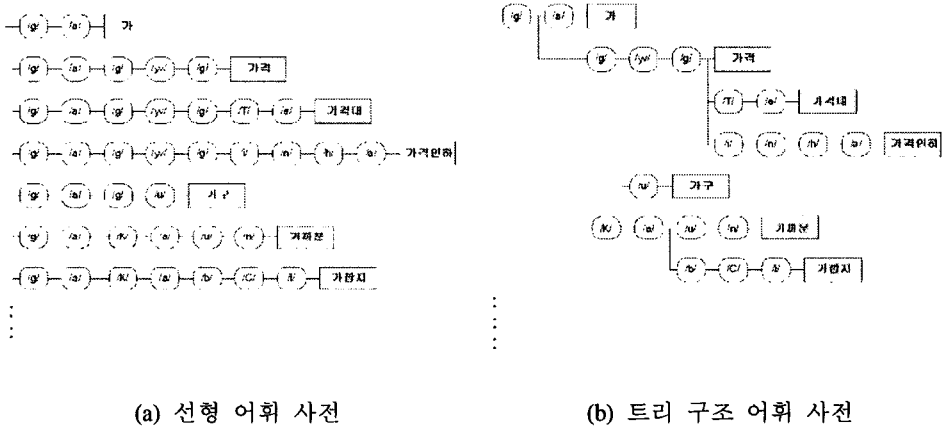
본 연구에서는 구현 시 많은 문제점을 내포하지만, 탐색 공간을 축소시켜 인식 시간을 빠르게 하는 트리 구조의 어휘 사전을 이용하여 연결 숫자음 인식 시스템을 구현하고, 구현 시 고려할 사항에 대해 언급한다. 본 논문의 구성은 다음과 같다. 먼저 2장에서는 연결 숫자음 인식 시스템의 기본 요소를 살펴보고, 3장에서는 트리 구조의 어휘 사전을 이용하여 인식 시스템의 구현 시 고려 사항을 지적하고, 기존 시스템과 구현된 시스템의 성능을 비교한다. 마지막으로 4장에서 본 연구의 요약 및 결론을 맺도록 하겠다.

## 2. 연결 숫자음 인식 시스템의 구성

본 장에서는 음성 인식 시스템에서 널리 사용되는 어휘 사전 모델링 방식을 설명하고, 대용량 음성 인식 시스템이나 비슷한 어휘를 가지는 음성 인식 시스템에서 인식 속도를 빠르게 하기 위하여 탐색 공간(search space)을 줄이는 트리 구조 사전 모델을 설명한다. 또한, 일반 음성 인식 시스템에서 널리 사용되는 토큰 전달 알고리즘을 설명하고, 트리 구조 어휘 사전을 이용하여 연결 숫자음 인식 시스템을 구현하도록 유한 상태 네트워크(FSN; Finite State Network)의 결합에 대해 설명한다.

### 2.1 어휘 사전 모델

대부분의 대용량 연속 음성 인식 시스템에서는 기본 인식 단위로 단어보다는 subword를 사용한다. 기본 인식 단위로는 주로 음소나 음소와 유사한 단위들을 주로 사용하게 되며, 이러한 음성인식 시스템에서는 단어나 문장을 인식하기 위해서 기본 subword 단위들로 구성된 단어 발음 사전(pronunciation lexicon)을 필요로 하게 된다. 단어 발음 사전은 일반적으로 단어에 해당하는 표준 발음 표기(baseform)를 인식 단위의 열로 나열함으로써 구성하거나 음운학적 지식을 가진 전문가에 의해서 만들어진다.



<그림 1> 발음 사전의 구조

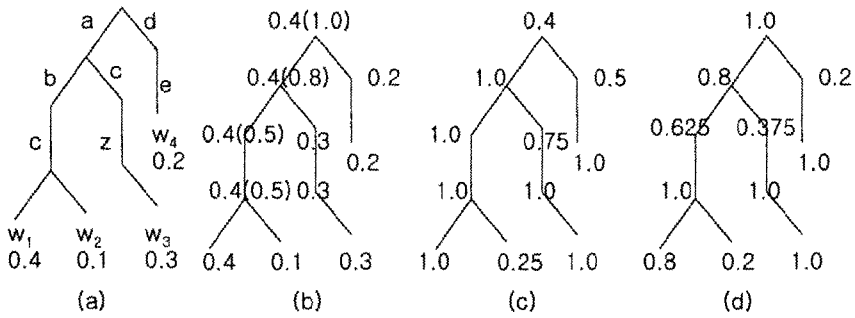
발음 사전은 <그림 1>에서 보듯이 선형 어휘 사전 형태나 트리 구조 어휘 사전 형태로 구성이 가능하다. 선형 어휘 사전 형태는 기본적인 발음 사전의 구조로 각 음소들이 일렬로 나열된 형태이다. 이 구조에서는 인식에 사용되는 네트워크(network) 크기가 발음 사전에 나타나는 전체 음소 수와 일치하게 된다. 그러나, 실제로 발음 표기 열들을 관찰해 보면 같은 음소 열들이 많이 중복되어 있다는 것을 알 수 있다. 이러한 중복된 음소 열들을 제거하거나 서로 공유하게 되면 네트워크의 크기가 줄어들고 탐색 시간을 줄일 수 있게 된다. 음소열의 중복은 대상 어휘에 따라 정도가 다르나, 일반적으로 어휘가 늘어날수록 중복되는 부분이 많아지게 된다. 트리 구조의 사전 형태에서는 여러 단어가 중복된 음소열 중에서도 초기의 같은 음소 열들을 공유하는 구조로 되어 있다. 네트워크의 크기는 트리의 노드(node) 수와 일치하게 되므로, 선형 어휘 사전에 비해서 탐색 공간을 많이 줄일 수 있게 된다. 그러나 트리의 마지막 노드에 도달하기 전까지는 인식 대상 단어(target word)를 결정할 수 없기 때문에 탐색 알고리즘에 언어 모델을 적용하기가

쉽지 않다는 문제점이 있다. 현재 많은 대 어휘 연속 음성 인식기에서는 인식 네트워크의 크기를 줄임으로써 탐색의 시간과 beam 탐색에서 발생하는 오류를 줄일 수 있기 때문에 트리 구조 형태의 발음 사전을 주로 사용하고 있다. 트리 구조의 어휘 사전을 이용하는 경우, 실제로 구현상의 문제점들이 보고되고 있지만, 대용량 음성을 실시간으로 인식하기 위해서는 트리 구조의 어휘 사전이 필요하다 [1].

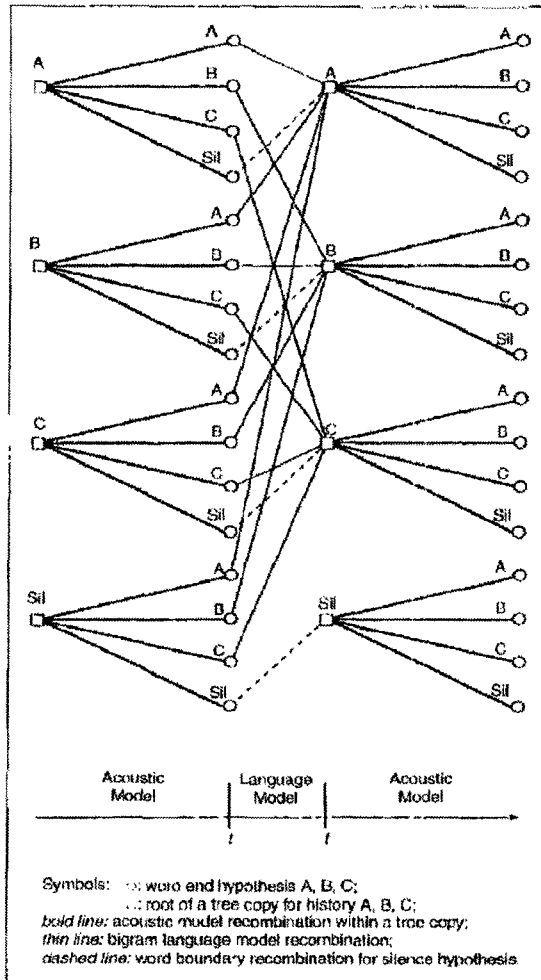
연결 숫자음 인식에 사용하는 어휘 모델의 경우, 일반적으로 선형 구조의 발음 사전을 많이 사용하고 있다. 그러나 모노폰이나 단어 모델이 아닌 트라이폰을 인식 대상으로 하는 경우, 단어의 처음에 오는 음소와 단어의 끝에 오는 음소에 영향을 받는 것을 제외하면, 대부분의 음소가 공유된다는 특징이 존재한다. 이러한 이유로 일반적으로 트라이폰을 인식 단위로 하는 경우, 접두어 공유 트리 구조(prefix sharing tree structure)나 접미어 공유 트리 구조(suffix sharing tree structure)의 어휘 트리(lexical tree)를 이용하고 있다. 본 연구에서는 기존의 시스템을 접두어 공유 트리 구조의 발음 사전을 이용하도록 개선하였다. 그러나, 트리 구조의 발음 사전을 이용하는 경우, 크게 두 가지 정도의 고려사항이 발생한다. 첫째는 트리 구조의 발음 사전의 경우, 트리의 단말노드에 도착하기 전까지는 인식 대상을 파악하기 어렵다는 단점이 있다. 이러한 문제점을 해결하기 위하여 단어의 단일 발생 확률을 tree의 각 줄기(arc)에 배분한 factored language probability를 적용한다. 일반적으로 word unigram probability를 적용하여 트리 구조의 발음 사전을 탐색하는데 가지치기(pruning) 하거나, word bigram probability를 적용하기도 한다. 이와 같이 tree 구조의 발음 사전에서 단어의 관측 확률을 이용하여 탐색하는 공간을 제약하는 방법을 LM (Language Model) Look-Ahead라고 한다. 본 연구에서 적용하는 각 단어의 출현 확률을 tree 전체로 분해하는 방법은 다음과 같다.

- ① 각 단어의 출현 확률을 tree의 단말에 지정한다.
- ② 트리 구조의 발음 사전에서 자식 노드(children)의 지정된 확률 값의 최대 값 또는 합을 현 노드의 확률 값으로 지정한다.
- ③ 모든 노드에 대해 확률 값이 지정되면, 자식 노드의 확률 값을 현재 노드의 확률 값으로 나눈다.

기존 연구에서는 ②단계에서 확률 값의 최대 또는 합을 적용하더라도 인식 성능에는 큰 차이가 없는 것으로 발표되었다[1].



<그림 2> (a) 분해되지 않은 발음 사전 (b) MAX 확률의 분산 () 안의 값은 확률 합(SUM)을 이용 (c) 최대(MAX) 확률 값을 이용한 분해 (d) 확률 합(SUM)을 이용한 분해



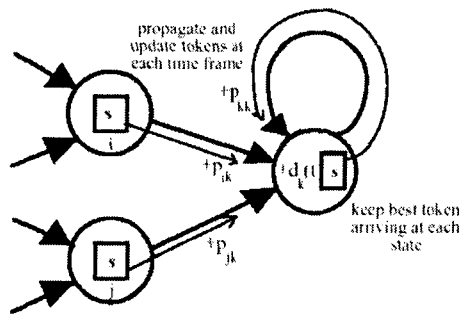
<그림 3> bigram 언어 모델을 이용한 경우의, 트리 구조의 발음 사전의 표현 [2]

그러나 숫자음 인식과 같이 단어간의 연결 정보를 확률로써 표현하기 힘든 경우에는 LM Look-Ahead 기법을 사용할 수 없다는 단점이 존재한다.

또 다른 고려 점으로는 트리 구조의 발음 사전을 이용하여 탐색 과정을 거치는 경우, 발음 사전의 복사로 인한 메모리의 증가와 탐색 알고리즘의 복잡성이다. 일반적으로 언어 모델을 사용하지 않는 경우, 한 단어 다음에 올 수 있는 단어는 모든 단어가 될 수 있기 때문에, 이전 단어에 대한 정보를 저장하지 않아도 된다. 그러나 bigram 이상의 단어 관측 확률을 이용하는 경우, tree 구조의 탐색 공간에서 결정된 현재의 단어  $w$  다음에 올 수 있는 단어  $v$ 를 알고 있어야 하나, 트리 발음 사전의 단말 노드에 도착할 때까지 단어  $v$ 에 대한 결정이 유보되어 모든 트리 구조의 복사가 병행되어야 한다(<그림 3> 참조). 이 문제점을 해결하기 위하여 어휘 트리를 복사하는 대신, 어휘 트리의 정보를 매 프레임마다 복사, 전달하는 방식을 사용한다. 이렇게 매 프레임에서 인식 정보를 저장하여 전달하는 방식을 토큰(token) 전달 방식이라 한다.

## 2.2 토큰 전달 알고리즘

본 연구에서는 어휘 트리 구조의 발음 사전을 복사하는 대신, 하나의 어휘 트리 발음 사전을 저장하고, 어휘 트리 발음 사전을 따라, 탐색을 진행하는 토큰 개념을 도입하기로 하였다, <그림 4>와 <표 1>은 상태간의 전이에 따른 간단한 토큰 전달 알고리즘(token passing algorithm)을 나타낸 것이다.



<그림 4> 간단한 token passing 알고리즘

단일 단어내의 상태를 가정하고 토큰을 전달하는 알고리즘은 다음과 같이 작동한다. 토큰은 경로가 분할되면 토큰의 복사가 이뤄지고, 경로가 결합되면 최대 확률 값 또는 최소 경로 값을 갖는 토큰만 유지된다. 따라서 토큰 전달 알고리즘은 일 단계 동적 정합(one-pass dynamic programming) 알고리즘과 동일하게 적용될 수 있다.

트리 구조의 어휘 사전을 이용하는 경우, 토큰의 복사는 인식 단위 모델 내부

에서 상태간의 전이가 발생하는 경우와, 단위 모델을 공유하는 모델간의 연결에서 발생한다(<그림 5> 참조). 또한 어휘 트리의 단말 노드에 토큰이 도달하면 인식 단어를 결정하게 되므로 새로운 단어에 대한 전이가 필요하다. 어휘 트리를 이용하는 경우 인식 단어의 결정이 어휘 트리의 단말노드에 이르기까지 유보되기 때문에 인식 단어가 결정되어 다른 단어로 전이해야 하는 경우 무조건적으로 토큰의 복사가 이루어진다[3].

<표 1> 동일 단어에서의 간단한 토큰 전달 알고리즘

**Initialization:**

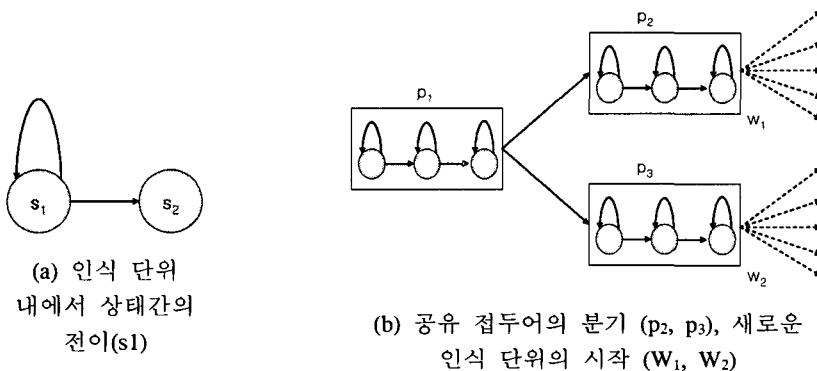
Each model initial state holds a token with value  
All other states hold a token with value

**Algorithm:**

for t = 1 to T do  
for each state i do  
Pass a copy of the token in state i to all connecting  
states j, incrementing its s value by  $p_{ij} + d_j(t)$   
end  
Discard the original tokens  
for each state i do  
find the token in state i with the smallest s  
value and discard the rest  
end  
end

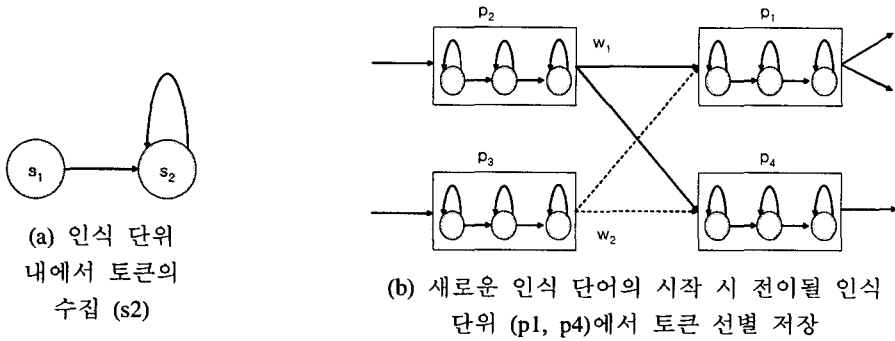
**Termination:**

Examine all final states, the token with the smallest s value  
gives the required minimum matching score



<그림 5> 토큰의 복사가 진행되는 경우

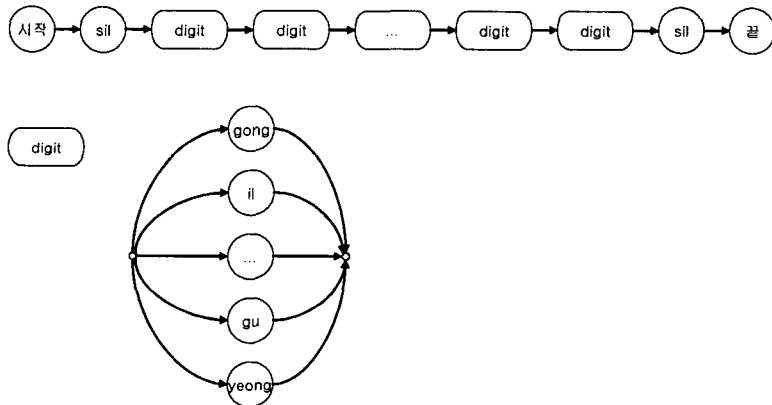
어휘 트리에서 토큰이 수집되어 최대 확률 또는 최소 경로의 토큰만 유지되는 경우는 인식 단위 내에서는 한 상태에 여러 경로가 결집되는 경우와 단어의 끝 상태까지 전이가 진행되어 동일한 프레임에서 새로운 단어가 시작되는 경우이다 (<그림 6> 참조). 인식 단어가 결정되어 새로운 단어가 시작되는 경우, 무조건적인 토큰의 복사가 이뤄지는데 전이될 인식 단위의 처음 상태에서는 최대 확률 값이나 최소 경로 값을 갖는 토큰만 저장하고 나머지는 버린다.



<그림 6> 토큰의 결집이 발생하여 한 토큰만 선별적으로 저장

### 2.3 어휘 트리와 유한 상태 네트워크 결합

고정 길이의 연결 숫자음 인식을 위해서 연결 단어를 유한 상태 네트워크로 표현하여야 한다. FSN을 표현하는 방법에는 여러 가지가 있을 수 있으나, 본 연구에서는 HTK(Hidden Markov Toolkit)[4]에서 사용하고 있는 단어 네트워크(Word Network)의 형식을 따라 유한 상태 네트워크를 구현하였다.



<그림 7> 8연결 숫자음의 유한 상태 네트워크



단어 네트워크는 노드(node) 리스트와 아크(arc, 호) 리스트로 구성되어 있다. 노드는 단어를 표현하고 아크는 단어 간의 전이(transition)를 의미한다(<그림 7> 참조). HTK의 표준 격자 형식(SLF; Standard Lattice Format)에서 노드와 아크의 정의는 줄 단위로 표현되며, 여러 필드로 구성된다. 단어 단위의 SLF 형식을 기술하는 것이 어렵지 않더라도 시간이 많이 걸리는 작업이기 때문에, HTK에서 제공하는 HParse를 이용하여 확장 BNF(extended Backus-Naur Form)에서 간단하게 SLF 형식을 구할 수 있다.

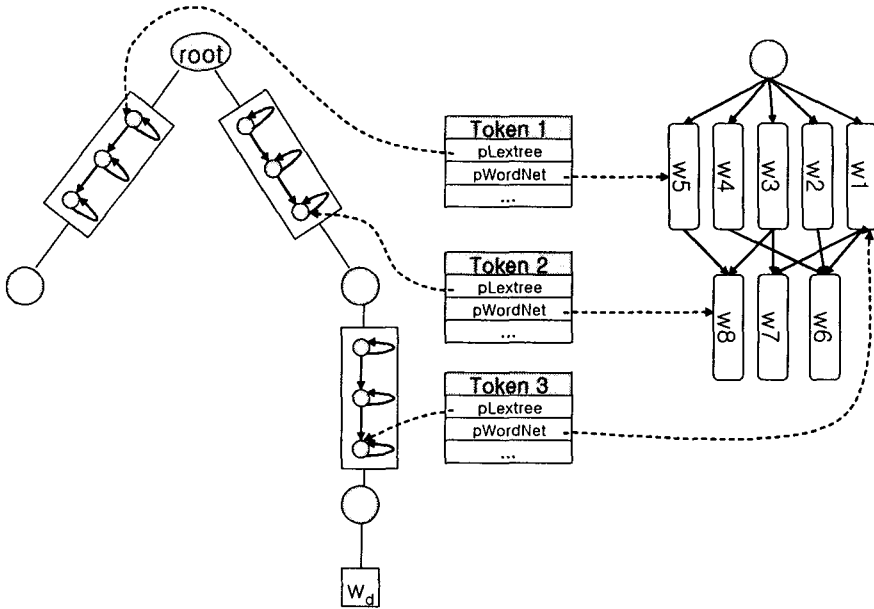
일반적으로 동일한 단어의 반복이 많은 경우에는 단어 네트워크를 공유할 수 있기 때문에 부분 네트워크(sub network) 개념을 도입하여 SLF 형식을 정의할 수 있다. 이 경우, 부분 네트워크를 먼저 기술한 다음, 다음 단계에서 부분 네트워크를 이용하여 다시 노드를 구성하여야 한다.

인식 단계에서는 이 SLF 파일을 읽어서 단어 네트워크를 구성하고, 어휘 트리의 단말 노드에서 인식 단어가 결정되면 저장된 단어 네트워크의 노드에서부터 전이 가능한 단어인지를 검사하여 부적합한 단어인 경우 버리고, 적합한 단어의 경우는 인식된 단어를 저장하고 탐색을 계속한다.

기존의 단어 네트워크와 일단계 정합 알고리즘의 결합은 단어의 인식이 끝난 후, 새로운 단어로의 전이과정에서 단어 네트워크를 이용하여 탐색 범위를 제한하였다. 그러나 어휘 사전을 이용하는 경우, 현재 인식중인 단어는 어휘 트리의 단말노드에 도달할 때까지는 결정되지 않기 때문에 유한 상태 네트워크가 탐색 범위를 제한하는 목적으로 사용될 수 없다.

따라서, 어휘 사전을 이용한 인식 시스템을 구성하는 경우 전절에서 설명하는 바와 같이 토큰 전달 알고리즘을 이용하여 구현할 수 있다. 이 경우, 각 토큰에는 대응되는 어휘 트리의 노드와 상태 번호 등이 저장된다. 이 토큰 알고리즘과 유한 상태 네트워크를 결합하기 위해서는 각 토큰에 단어 네트워크의 노드에 대응되는 지시자(pointer)가 저장되어야 한다(<그림 8> 참조). 이 지시자는 어휘 트리의 단말 노드에서 인식 단어가 결정되면 이전 단어에서 전이 가능한 단어인지를 검사하는 역할을 한다.

<그림 8>에서 토큰 1의 경우, 단어를 구성하는 인식 단위 내에서 전이가 발생하는 경우이기 때문에, 어휘 사전의 각 인식 단위를 가리키는 지시자는 유지되고, 상태 정보만 수정된다. 토큰 2의 경우에는 인식 단위가 바뀌는 경우로서 어휘 트리의 노드 정보와 상태 정보가 수정된다. 마지막으로 토큰 3의 경우, 인식 단어가 결정되었기 때문에 현재 토큰에 저장된 단어 네트워크의 노드 정보를 이용하여 인식된 단어가 전이 될 수 있는지를 검사하여 전이 가능하지 않으면 폐기 처분한다. 즉, wd가 저장된 w1의 다음어들 중의 하나인 w6, w7 중의 하나이면 현재 단어 네트워크의 노드 정보를 갱신하여 토큰을 저장하고, 그렇지 않은 경우 토큰 3은 폐기처분되어 더 이상의 탐색이 진행되지 않는다.



<그림 8> 어휘 트리과 단어 네트워크, 토큰의 상관관계

이 과정을 정리하면 <표 2>와 같이 간단하게 기술할 수 있다.

<표 2> 어휘 트리에서의 토큰 전달 알고리즘과 FSN의 결합 알고리즘

```

Initialization: t = 0
Find the available start word and set the previous word network
pointer to NULL
Algorithm: Token propagation
for t = 1 to T do
  for each state i do
    Pass a copy of the token in state i to all connecting
      states j
    If the state i is the end state of the subword unit,
      new token is generated with next subword unit
      information
    else if the state i is the end state of the target word,
      check whether the current word is able to be
      propagated from the saved word node of network
      end
    end
  end
end
    
```

```

Discard the original tokens
for each state i do
find the token in state i with the maximum
likelihood and discard the rest
end
end
Termination:
Examine all final tokens, find the token of which
the corresponding state is at the end state of the sentence
end word and report the path

```

제안된 알고리즘의 문제점은 입력 음성이 종결되더라도 단어 네트워크의 종료 노드에 지시자가 도착하지 못하는 경우이다(탐색 오류). 이것은 인식 단위가 음향학적으로 제대로 모델링되지 못한 경우 발생하는 현상으로 특정 단어가 입력 음성에 지배적으로 반응하는 경우이다. 즉, 각 상태에서 탐색 시점에서 최대 확률을 갖는 토큰만 저장되기 때문에 발생하는 지역 최적화 문제(local optimum)로 해석된다. 이 현상은 일반적인 OPDP 알고리즘의 단점을 그대로 반영하고 있기 때문에 정확한 음향 모델링을 통해서 해결될 수 있거나 트리 구조의 어휘 사전과 단어 네트워크를 결합하는 방식에 의하여 해결될 수 있다.

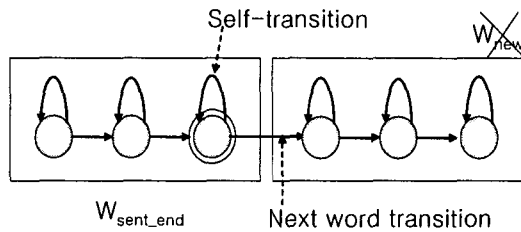
### 3. 구현 및 실험 결과

본 장에서는 트리 구조의 어휘 모델링을 적용하기 위해 연결 숫자음 인식 시스템의 구현 과정에서 고려할 사항과, 구현된 인식 시스템의 성능을 기존의 방법과 비교하여 인식 실험을 하고 그 결과를 정리한다.

#### 3.1 구현 시 고려할 점

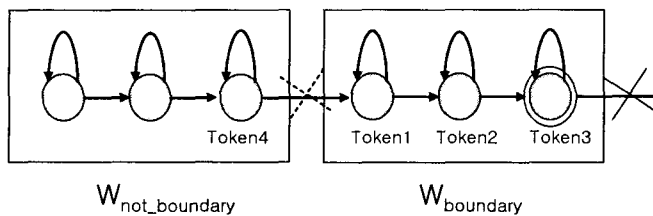
어휘 트리를 적용한 연결 숫자음 인식 시스템을 구현하는 과정에서 탐색 시 발생 가능한 상황을 언급하고 해결 방안을 제시한다. 어휘 트리를 이용한 탐색 알고리즘인 경우, 인식되는 단어는 어휘 트리의 단말 노드에서 결정된다. 이처럼 지연된 인식 단어의 결정은 단어 네트워크를 이용하여 유한 상태 네트워크를 구현하였을 경우, 문장의 끝에서도 계속해서 새로운 단어가 전이될 수 있다(<그림 9> 참조). 따라서 각 인식단어가 결정된 후 단어 네트워크에서 문장의 끝 단어인지를 검사하여 더 이상의 새로운 단어로의 전이를 금지하도록 한다. 그렇지 않고, 새로

운 단어로의 전이 확률과 인식 단어( $W_{sent\_end}$ )의 자체 전이(self-transition)의 확률을 비교하여 새로운 단어로의 전이 확률이 더 커지게 되거나, beam pruning에 의해 자체 전이 경로가 삭제되거나, 입력된 음성의 마지막 프레임에서 인식된 단어( $W_{new}$ )가 문법에 의해 거부되는 경우 제대로 된 문장을 인식할 수 없다. 따라서 각 인식된 단어를 검사하여 문장에서의 마지막 단어인 경우, 새로운 단어( $W_{new}$ )로의 전이 경로를 금지한다.



<그림 9> 문장의 끝 단어인 경우 새로운 단어로의 전이 금지

탐색과정에서 발생할 수 있는 다른 하나의 문제점은 문장의 끝에 존재하는 인식 단위(예; sil)에서 부적합한 단어의 전이로 인하여 저장된 토큰의 정보가 삭제되는 경우이다. 본 연구에서는 효율적인 메모리 운영을 위하여 단일의 어휘 트리와 각 프레임별 가능한 토큰을 이용하여 언어 정보나 유한 상태 네트워크를 표현하고 있다. 따라서, 전이 가능한 어휘 트리의 모든 상태는 단일의 토큰 정보와 연관지어진다. 그러나, 부적합한 단어 전이는 어휘 트리의 단말노드에서 인식 단위의 마지막 상태에 도달해서만 파악할 수 있기 때문에, 적합한 토큰 정보를 부적합한 토큰 정보(마지막 상태에서 폐기될 토큰)가 겹쳐쓸 수 있게 된다. 모든 인식 단위의 부적합한 전이를 파악하기는 어렵기 때문에 단어의 경계를 표현하는 sil 단어는 미리 탐색 과정에서 방지하는 것이 인식 성능을 향상시키는데 도움이 될 것이다 (<그림 10>).



<그림 10> 경계 단어로의 부적합한 전이 금지

위 그림에서 경계 단어( $W_{boundary}$ )는 문장의 끝에 도달되어 더 이상의 전이가 금지되어, Token3는 계속 자체 전이만 허용된다. 이때 새롭게 결정된 인식 단어

$W_{not\_boundary}$ 는 다음에 전이될 단어를 모르기 때문에 모든 가능한 인식 단위로의 전이가 허용되어 Token4의 정보가 최대 값 유지(maximum token kept) 정책으로 인하여  $W_{boundary}$ 의 마지막 상태에까지 도달할 수 있게 된다. 그러나  $W_{boundary}$ 의 마지막 상태에서 부적합한 단어의 전이로 판정되어 그 경로는 버려진다. 이런 경우, 적합한 경로 정보를 저장하고 있는 Token1, Token2, Token3는 이미 Token4에 의해 갱신된 경우이기 때문에 인식 오류를 발생하게 된다. 따라서, Token4가 새로운 단어로 전이하는 경우에 경계 단어인  $W_{boundary}$ 로 전이를 미리 검사하여 부적합한 경로 생성을 줄인다면 인식 성능은 향상될 것이다.

### 3.2 비교 평가

실험은 기존 연구와 동일한 환경에서 이루어졌다[5]. 기존 연구에서 음소 모델이나 단어 모델인 경우 단어 별 상태 수에 따라 성능의 변화가 커, 본 연구에서는 평균 단어 길이에 기초해 음소 모델의 상태 수를 다르게 적용한 결과 성능의 변화가 급격하다는 것을 알 수 있었다. 기존의 연구와 동일한 환경에서 실험한 결과 매 프레임에서 분기 가능한 최대 토큰의 수(인식 어휘의 총 상태 수)는 <표 3>과 같다. 표에서 알 수 있듯이 트리 구조를 채택한 어휘 사전이 약 60% 정도의 메모리를 절약할 수 있었다. 특히, 본 연구에서는 단어간 트라이폰을 고려하였기 때문에 메모리를 많이 절약할 수 있었다.

<표 3> 선형 어휘 모델과 트리 구조 어휘 모델의 상태 수 비교

어휘 사전 모델	선형 모델	트리 구조 모델
총 상태 수	19,721	7,980

삼음소 모델 기반의 인 환경에서 인식 실험한 결과는 <표 4>에 정리되었다. 이 실험에서는 삼음소 모델을 적용하여 '이'와 '오'에 해당하는 음소는 상태 수를 6개로 결정하고 나머지 음소에 대해서는 4개의 상태를 할당하였다. 실험 결과 단어 오류율에서는 9.26%, 문장 오류율에서는 8.8%, 인식 시간에서는 64.3%의 성능 향상을 이루었다.

<표 4> 구현된 시스템의 성능 향상

	선형 모델	트리 구조 모델	성능 향상
인식 성능 (WER)	5.4%	4.9%	9.26%
인식 성능 (문장)	71.6%	74.1%	8.8%
평균 인식 시간	986.56sec	352.22sec	64.3%

#### 4. 요약 및 결론

본 연구에서는 효율적인 연결 숫자음 인식 시스템을 개발하기 위하여 트리구조의 어휘 모델과 유한 상태 네트워크를 결합한 탐색 알고리즘을 구현하였다. 구현된 알고리즘은 어휘 트리를 이용하여 통계적 언어 모델을 이용한 연속 음성 인식 및 제한된 문법을 허용하는 유한 상태 네트워크를 모두 허용할 수 있도록 한 개의 저장된 어휘 트리 와 토큰 전달 알고리즘을 이용하였다. 특히, 어휘 트리 와 유한 상태 네트워크를 동시에 만족하도록 하는 경우 발생할 수 있는 문제점을 지적하고, 그 해결책을 제안하였다.

구현된 시스템의 효율성을 확인하기 위하여 기존의 연구와 동일한 환경에서 실험한 결과 인식률이나 속도, 메모리 면에서 모두 괄목할 만한 성능향상을 보였다. 특히 매 프레임에서 전이 가능한 총 상태 수의 감소를 통하여 인식 시간을 1/3 수준으로 속도를 향상시켰다. 현재 구현된 어휘 트리는 접두어만을 공유하는 구조이나 추후 연구로 접두어와 접미어를 모두 공유하는 어휘 모델에 관한 연구를 통하여 더 나은 인식 성능과 속도 향상을 꾀하고자 한다.

#### 참고 문헌

- [1] X. Huang, A. Acero, H.-W. Hon, "Spoken Language Processing - A Guide to Theory, Algorithm, and System Development", pp. 645-682, Prentice Hall, 2001
- [2] H.Ney, "Dynamic Programming Search for Continuous Speech Recognition", IEEE Signal Processing Magazine, pp.64-83, 1999
- [3] S.J. Young, N.H. Russell, J.H.S Thornton, "Token Passing: a Simple Conceptual Model for Connected Speech Recognition Systems", Technical Report CUED/F-INFENG/TR38, Cambridge University Engineering Department, 1989
- [4] S.J. Young et al., "The HTK Book", Cambridge University, 2001
- [5] 윤영선, 박운상, 채의근, "연결 숫자음 인식 시스템의 구현과 성능 변화", 말소리, No. 45, pp.47-61, 2003

접수일자: 2004년 5월 11일

게재결정: 2004년 6월 7일

▶ 운영선 (Young-Sun Yun)

주소: 대전시 대덕구 오정동 133번지

소속: 한남대학교 정보통신공학과

전화: 042) 629-7569

Fax: 042) 629-7843

E-mail: ysyun@hannam.ac.kr

▶ 채의근 (Yi-Geun Chae)

주소: 충남 공주시 옥룡동 326

소속: 공주대학교 멀티미디어정보·영상공학부

전화: 041) 850-6210

Fax: 041) 858-0580

E-mail: ygchae@kongju.ac.kr