반음절기반의 한국어 연속숫자음인식과 그 후처리에 대한 연구

A Study on Korean Connected Digit Recognizer Based on Semi-syllable and Post-processing

정 재 부* · 정 훈** · 정 익 주***
Jae-boo Jeong · Hoon Chung · Ik-joo Chung

ABSTRACT

This paper describes the effect of new recognition unit, a unit based on semi-syllable, and its post processing method. A recognition unit based on semi-syllable expresses Korean connected digit's coarticulation effect. An existing method using semi-syllable limits next models, derived from current recognized models, to make complete connected digit sequence. However, this paper uses a new method to make complete connected digit sequence. The new post-processing method recognizes isolated digit words which include digits sequence from the digit combinations being able to occur from current recognized semi-syllable sequence. This method gives an improved accuracy rate than that of existing method. This new post processing provides two advantages. 1) It corrects current mis-recognized semi-syllable unit. 2) When people say each digit, they say it without regard to saying duration.

Keywords: Continuous Digit Recognition, Semi-syllable

1. 서 론

인간은 기계장치를 사용하기 시작하면서부터 기계를 제어하는 수단으로 버튼이나 스위치, 키보드 같은 기계적인 입력장치를 사용하였다. 간단한 스위치부터 현재의 컴퓨터에서 사용되는 마우스, 키보드까지 인간의 손으로 기계를 동작시킴으로써 인간과 기계장치간의 의사소통(제어명령)이 이루어지고 있다. 하지만 이러한 의사소통장치는 인간의 기본적인 의사 소통장치격인 음성에 비해 많은 불편한 면을 가지고 있다. 이에 수십 년 간 음성으로 기계장치를 제어하려고 많은 연구가 이루어졌다. 현재의 음성인식 수준은 PC상에서는 수 만 단어인식이 가능하며, 영어의 경우 텍스트를 받아 적는 단계까지 발전하였다. 본 논문에서는 음성인식 중에서 숫자음에 대한 인식에 대해 논하고자 한다. 현재 사람들이 자신을 나타내는 수단으로

* 강원대학교 전자공학과 대학원
** 강원대학교 전자공학과 대학원
*** 강원대학교 전기전자공학부 교수

가장 많이 사용되는 것은 숫자이다. 주민등록번호, 은행계좌번호, 전화번호, 신용카드번호 등 다른 사람과 자신을 구분하는 것으로 숫자가 많이 사용되어 지고 있다. 영어의 경우는 대부분 숫자가 단음절이 아니고, 서로간의 연음현상이 적은 편이라 연속 숫자음에 대한 인식이 잘 이루어지고 있으나, 한국어의 경우는 모든 숫자음이 단음절이고, 'ㅣ' 모음으로 시작하는 단어들이 많아, 연음현상이 많이 발생하여 인식에 많은 오류가 생긴다. 한국어 연속숫자음의 경우 서로간의 연음현상에 대한 특징을 잘 반영한다면, 좋은 인식결과를 보일 수 있기에 본 논문에서는 이를 해결할 수 있는 새로운 방법인 반음절+반음절로 인식유닛을 생성하고[1], 이에 대한 후처리를 통해서 인식률을 향상시켰다.

본 논문에서는 필터뱅크분석모델을 기본으로 하는 MFCC(Mel-scale Frequency Cepstral Coefficient)로 특징벡터를 추출하고[2], 인식알고리즘은 Continuous HMM[3]을 사용한다. HMM에서 음소단위(mono-phone, di-phone 등)로 숫자음을 인식할 경우 연속적으로 발음된 숫자음의 인식시 정확하게 음소단위로의 분할이 이루어지지 않아 오인식이 많이 이루어지지만[4], 반음절+반음절로 인식유닛을 생성할 경우 연속적으로 발음된 숫자들간의 연음현상이 인식유닛상에 표현되기에 연음현상에 의한 오인식을 보정할 수가 있다. 그리고 인식되어진 반음절+반음절의 열을 다시금 재구성하여 재인식함으로써 좀더 정확한 인식률을 얻는 방법에 대해 논하고자 한다.

2. 숫자음 인식시스템 구현

2.1 반음절 기반의 인식 유닛

현재의 많은 음성인식시스템에서는 인식률 향상을 위해 음소 간의 연관성을 고려하여 mono-phone보다는 di-phone나 tri-phone을 사용하고 있다[5][6]. 하지만 서론에서도 언급되었듯이 한국어 숫자음의 경우 단음절로 이루어져 있고, 연음현상의 발생 경우가 많이 존재하여 인식률이 저하된다. 본 논문에서는 기존의 음소단위가 아닌 반음절로 인식단위를 변경하는 방법이 연음현상에 대해 잘 표현해 줄 수 있음을 확인하여[1] 이를 숫자음인식시스템에 적용하고, 후처리 과정을 거침으로써 숫자음인식시스템의 성능의 변화를 보고자 한다.

한국 숫자음은 '일', '이' 처럼 모두 단음절로 이루어져 있다. 이를 반음절로 구분 짓게 되면, '일'은 '이'와 'ㄹ'로 되며, '이'의 경우는 '이'와 'ㅣ'가 된다.

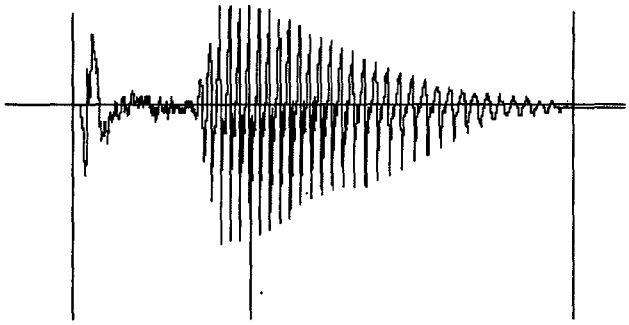

그림 1. '팔'을 반음절로 구분한 모습

그림 1은 숫자 '팔'을 반음절로 분할한 모습이다. 모음 'ㅏ'를 중심으로 앞의 반음절 '파'와 뒤의 반음절 '알'로 분할하게 된다.

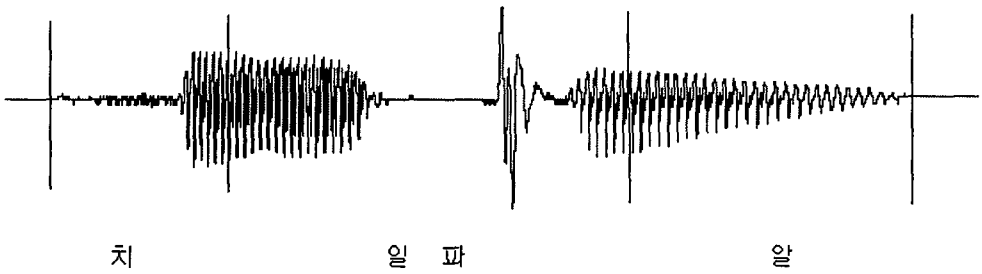

그림 2. '칠팔'을 반음절 단위로 나눈 모습

그림 2는 숫자 '칠팔'을 반음절로 분할한 모습이다. '칠팔'은 두 음절로 이루어졌기 때문에 '치', '일' 과 '파', '알'로 구분 지을 수 있다. 하지만 본 논문에서는 연속된 숫자의 연음현상을 모델에서 잘 표현하기 위하여 앞 음절의 '일'과 뒤 음절의 '파'를 반음절+반음절인 하나의 유닛으로 여긴다. 따라서 반음절로만 이루어진 부분('치', '알' 부분)과 반음절+반음절('일_파' 부분) 부분으로 나눌 수 있다. 이 중에서 '일_파'처럼 반음절+반음절로 된 부분에 해당하는 모델이 잘 형성되었을 경우 숫자음 인식률이 향상되게 된다. 단지 반음절('치', '일', '파', '알')을 연속적으로 사용하게 된다면 연속되는 숫자 사이의 연음현상에 대한 표현을 할 수 있는 모델이 없기에 음소를 인식단위로 하는 것과 차이가 없어진다. 그리하여 반음절을 인식의 기본 유닛으로 사용할 경우는 연속된 숫자음의 연음현상을 모델에 반영하기 위해 숫자의 처음과 끝 부분을 제외한 부분은 반음절+반음절을 기반으로 하여 모델을 형성하여 사용하게 된다.

본 논문에서는 전부 12 개의 숫자음[1]을 사용하였고, 각 숫자음들을 두 자리로 만들어 반음

1) '영', '공', '일', '이', '삼', '사', '오', '육', '륙', '칠', '팔', '구'를 사용하여 전부 12 개의 숫자음

절+반음절에 해당하는 모델을 형성하였으므로, 전부 168[2] 개의 모델이 사용되어졌다.

표 1. 반음절로 나뉜 인식유닛

숫자음	인식유닛(앞)	인식유닛(뒤)
영	eo	eong
공	go	ong
일	i	il1
이	y	y2
삼	sa	am
사	sa2	a1
오	o	o5
육	eu	euk
륙	reu	euk2
칠	ci	il7
팔	pa	al
구	goo	oo

표 2. '일이', '삼사', '오육', '칠팔'에 대한 인식유닛

숫자음	반음절	반음절+반음절	반음절
일이	i	il1_y	y2
삼사	sa	am_sa2	a1
오육	o	o5_eu	euk
칠팔	ci	il7_pa	al

위의 표 1, 2를 통해서 반음절기반을 인식의 기본 유닛으로 하고 숫자음을 인식할 경우, 숫자음들이 어떻게 반음절 또는 반음절+반음절로 나뉘게 되는지를 볼 수 있다. 인식과정을 거쳐 나오게 된 결과는 그대로 사용하지 않고, 분리-재조합과정을 다시 거쳐서 올바른 숫자음 유닛으로 변경해야지만 한다. 연음현상을 고려한 반음절+반음절 유닛은 서로 다른 두 개의 숫자음에서 반음절씩을 가져와 조합한 것이기에 이를 다시 반음절 단위로 분리하고, 분리된 반음절 단위의 유닛들을 재조합하여 올바른 숫자음 유닛을 만들어 낸다.

에 대해 반음절 모델을 형성하게 된다.

2) 반음절의 개수는 12*2 = 24
반음절+반음절의 개수는 12*12 = 144
따라서, 모든 유닛의 수는 168 개가 형성되게 된다. 본 논문에서는 silence 모델을 추가적으로 사용하였다.

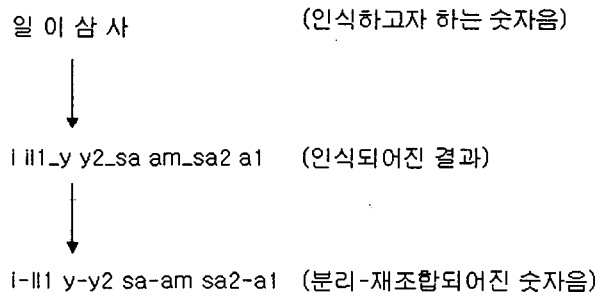

그림 3. 숫자음의 인식과 분리-재조합 과정

그림 3에서 보듯이 '일이삼사'라는 사연숫자의 인식결과는

i il1_y y2_sa am_sa2 a1

로 나오게 된다. 반음절 유닛인 'i' 와 'a1'을 제외한 나머지는 모두 앞뒤의 숫자음들의 반음절이므로

il1_y	→	il1	y
y2_sa	→	y2	sa
am_sa2	→	am	sa2

로 분리하여, 이를 정상적인 숫자음절에 해당하는 반음절들로 재조합하여 사용한다.

2.2 후처리 과정

2.1절에서 언급한 대로 반음절, 반음절+반음절로 인식유닛을 형성하여 인식을 하게 되면 연음현상을 잘 표현할 수 있기에 한국어 숫자음 인식에 적합하다[1]. 하지만 인식유닛을 반음절에 기반으로 하여 인식을 하더라도 반음절과 반음절이 이어지는 부분에서 서로 연계되는 정보가 없기에 유닛이 잘못 형성되는 경우가 발생한다. 예를 들어 '일이삼사'를 인식할 경우 올바른 반음절 단위로 분할이 되었다면,

일 이 삼 사 → i il1_y y2_sa am_sa2 a1

이처럼 분할되어 인식이 이루어져야만 한다. 하지만 'i' 이후에 'il1_y2'이 언제나 나온다는 보장이 없다[3]. 이것이 잘못 인식이 될 경우 '일이삼사'에서 '일이' 두 자리 숫자에 대한 인식이 잘못되기에 전체적인 숫자음에 대한 인식 결과가 그릇된 결과가 된다. 이처럼 잘못된 반음절들의 조합에 해당하는 문제를 해결하고자 본 논문에서는 인식되어진 결과를 다시 한번

3) 연속적인 숫자음 사이에는 어떠한 문법규칙이 존재하지 않는다. '일' 이후에 '이', '삼', '사' 등이 올 가능성은 모두 동일하기 때문이다.

별도의 인식과정을 거치게 하여 인식률을 향상시켰다.

기존에 제시된 반음절기반의 숫자음인식에서는 단지 앞에 인식된 반음절 중 가장 유사하다고 판단되는 일정의 유닛을 선택하여, 그와 연관된 반음절들을 다음의 인식모델 후보로 두어 인식에 들어갔다[1]. 그림 4에서 볼 수 있듯이 처음 반음절이 '0'이 인식되었다면 그 다음 인식되어질 모델은 01, 02 등 0으로 시작하는 반음절+반음절모델이 된다. 마지막 반음절 부분까지 모든 인식이 이루어지고 나면, 인식되어진 모델들을 나열하여 숫자열을 인식하게 된다. 그림 4에서 인식되어진 모델이 반음절 모델은 '0', 반음절+반음절모델은 '00', '09', 마지막 반음절 모델이 '9'라면, 인식되어진 숫자열은 '009'라는 삼연숫자가 되는 것이다. 이를 간략히 표현하면 다음과 같다.

0 00 09 9 → 0 0 9

이러한 후처리방법은 음성 입력시 주위의 잡음과 잘못된 발성에 의해 어느 시간의 반음절이 오인식되었을 경우 그 결과는 이후의 숫자인식에까지 영향을 미치게 된다. 이렇게 되면 뒷부분의 발성된 숫자음이 아무리 정확하게 발음되어진 것이라 하더라도 앞에서 인식된 반음절에 영향을 받아 오인식된 결과를 얻을 가능성이 있기에 본 논문에서는 이러한 후처리 방식을 채택하지 않고, 다른 방식을 채택하였다.

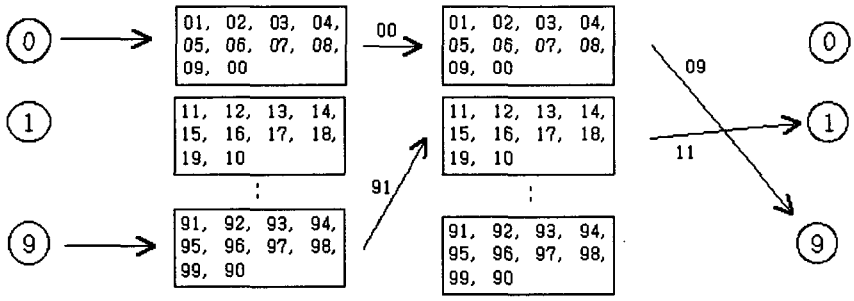

그림 4. 현재 인식되어진 모델에 따라 다음 인식모델을 제한하는 후처리

본 논문에서는 후처리를 위해 별도의 고립단어인식을 첨가하였다. 후처리를 하기 전에 인식되어진 모델은 기존에 제시되었던 방법 [1]과는 다르게 현재 인식되어진 인식모델이 다음의 인식모델을 제한하지 않고, 어떠한 모델이라도 인식모델이 될 수 있다. 후처리 과정 이전에 인식되어진 결과를 조합하였을 때, 조합된 반음절이 불완전한 조합[4]을 이루게 될 경우, 불완전한 조합에 대한 모든 가능성의 숫자음을 발생시켜, 이 모두를 고립단어 인식을 거치게 하여 최종 인식결과를 얻게 된다.

4) 예를 들어 '일'의 경우 'i'+'il1'의 유닛으로 올바르게 조합되어졌을 경우는 완전한 조합이라 하고, 이 이외의 경우는 불완전한 조합으로 본다.

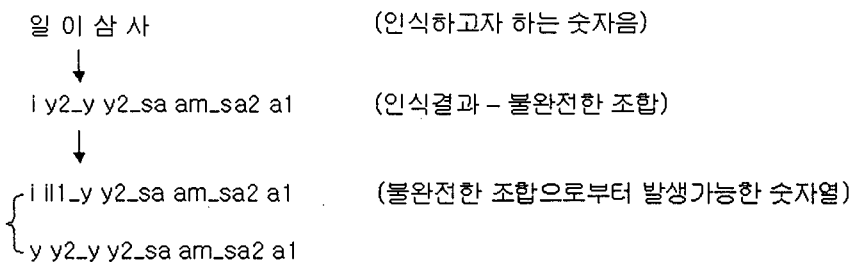

그림 5. 반음절을 이용한 인식과 후처리 과정

그림 5처럼 인식 결과가 불완전한 조합으로 발생하였을 경우 불완전한 조합인 'i'와 'y2'는 발생 가능한 숫자음인 '일'과 '이'로 분리되어 숫자음유닛을 재형성한다.

재형성한 이후에는 각각의 경우를 고립단어인식을 거쳐서 최종의 인식결과를 얻게 됨으로써 발성된 숫자음과 제일 유사한 결과를 얻을 수 있다. 후처리를 하는데 사용되는 고립단어 시스템은 기본적으로 본 논문이 숫자음인식시스템과 동일하다. 단, 문법(grammar)정보의 차이를 두어 인식방식을 구분하는 것이다. 숫자음의 경우는 자리수를 정해 놓으면, 각 자리수마다 모든 경우의 숫자음이 발생하도록 문법정보를 정의한다. 그래서, 인식유닛이 자유롭게 나올 수가 있는 것이다. 이러한 자유스러움이 연속적인 숫자음을 인식하려할 때 오인식의 원인이 된다. 고립단어의 경우는 불완전한 조합에서 새롭게 형성된 정상적인 숫자음들에 대해서만 연속적인 반음절기반의 인식모델을 형성하여 인식을 하게 되기에 자유로운 문법정보를 사용하는 숫자음의 경우보다 좋은 숫자음에 대한 인식률을 보인다.

후처리에 대한 인식률의 향상에 대해서는 다음 장에서 자세히 알아본다.

3. 실험 및 결과

숫자음인식시스템의 성능은 HMM 파라미터값을 어떻게 조정하여 사용하는가에 따라 영향을 받는다. 본 논문에서는 반음절기반 인식유닛에 다양한 state수를 적용하여 인식시스템의 성능을 테스트하였다.

본 논문의 인식시스템에서는 원광대학교에서 작성한 다양한 형태의 숫자음성 DB를 사용하였다. 이 중에서 단독숫자 12 개와 사연숫자 864 개[5], 다연숫자 1,440 개의 데이터에 대해 남자 200 명의 데이터로 학습을 시켰다. 그리고 인식률을 테스트하기 위한 데이터로는 사연숫자 777 개로 위의 200 명에 포함되지 않은 남자 10 명분의 데이터를 가지고 사용하였다.

3.1 state 수에 따른 인식률

본 논문에서는 반음절 기반의 인식유닛을 사용하여 인식시스템을 구현하였다. 인식시스템에 사용되어지는 각 인식유닛은 그것의 state 수를 어떻게 정하느냐에 따라 입력된 음성신호를 잘 표현할 수도 있고 그렇지 못할 수도 있다. 잘 표현되었을 경우는 좋은 인식률을 보이

5) 864 개의 사연숫자상에는 삼연숫자의 모든 경우가 포함되어 있다.

지만 그렇지 못한 경우는 나쁜 인식률을 보인다. 본 절에서는 반음절 기반으로 인식유닛을 사용하였을 때 어느 정도의 state를 사용하는 것이 좋은지를 테스트하였다.

본 논문에서는 다음과 같은 HMM 모형을 사용하고 있다.

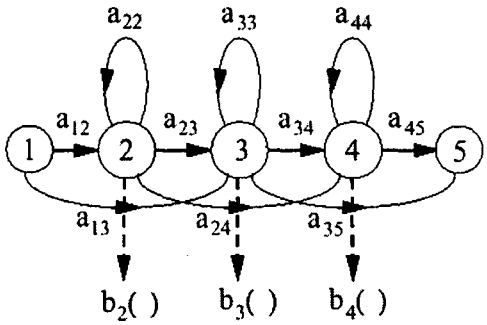

그림 6. state가 5인 HMM 모습

그림 6을 보면, state가 5인 HMM은 1 개의 entry state(state 1)와 1 개의 exit state (state 5), 그리고 3 개의 emitting state(state 2, 3, 4)로 이루어져 있다. 그림 5에서 1, 5 번의 state는 각각 entry, exit state로 non-emitting state이다. non-emitting state는 output probability distribution을 가지지 않는다. 단지 모델 간을 연결할 때 상태 변화에만 관여를 하는 state들이다. 따라서 실제 인식시 관측열에 관계된 state수는 entry와 exit를 제외한 state들의 개수이다.

3.1.1 후처리를 하지 않은 상태에서의 실험

2.2 절에서, 본 논문에서의 숫자음인식시스템은 반음절에 기반한 기본 인식유닛으로 인식을 한 후, 이를 통해 얻어진 결과를 가지고 다시 후처리를 거쳐 인식률을 향상시켰다고 언급하였다. 우선 후처리를 거치지 않았을 경우, state에 따라 인식률[6]의 변화를 알아보면 다음과 같다.

6) 본 논문에서는 state 변화에 따른 인식률을 두 가지로 나누었다. 한 가지는 자릿수에 따른 인식률이고, 다른 한 가지는 전체 숫자음(사연)의 인식률이다. 예를 들어, '일이삼사'를 인식하고자 할 때, 인식된 결과가 '일이삼사'였다면 자릿수에 따른 인식률은 100%, 전체 숫자음의 인식률도 100%이다. 하지만 인식된 결과가 '이이삼사'였다면, 자릿수에 따른 인식률은 75%이고, 전체 숫자음의 인식률은 0%이 되는 것이다.

표 4. state에 따른 후처리를 거치지 않은 인식률 (단위: %)

state 수	자릿수에 따른 인식률	전체 숫자인식률(사연)
5	86.36	55.98
10	92.05	71.30
11	92.89	74.65
12	92.95	75.03
13	92.63	74.00

표 4를 보면 state가 12일 때 가장 좋은 인식률을 보인다. state가 5일 경우는 반음절+반음절에 해당하는 부분의 상태 변화를 잘 표현하지 못하기에 가장 낮은 인식률을 보인다. 일반적으로 하나의 음소마다 5 개의 state를 설정하여 파라미터값을 계산한다. 그러므로 음소보다 상태변화가 많이 존재하는 반음절+반음절을 state 5로 하여 표현하고자 한다면 너무 적은 state에 의해 상태변화가 잘 표현되지 못하여 인식률의 저하를 보인다. state 13의 경우는 state가 많아짐으로써 반음절+반음절에 대한 상태변화에 대한 표현을 잘 할 수 있지만, 지나친 상태변화에 따라 오히려 인식률의 저하를 보인다. 표 4를 통해 알 수 있듯이 반음절+반음절의 경우는 state 12가 가장 적당하다. 반음절+반음절의 state가 12라면 반음절은 그보다 작아야하지만 지금의 테스트에서는 모든 인식유닛(반음절과 반음절+반음절)의 state를 동일하게 두고 테스트하였다. 인식유닛이 반음절인지, 반음절+반음절인지에 따라 state수의 변화를 주는 실험은 3.1.3 절에서 이루어진다. 그리고 불완전한 조합으로 인식 결과가 나왔을 경우, 이 실험에서는 후처리과정으로 고립단어인식을 하지 않았다. 따라서, 불완전한 조합의 인식결과가 발생하였을 경우 다음의 세 가지 규칙을 따라 인식결과를 표시하였다.

- 인식된 유닛이 o-oo일 경우는 '오'로 가정한다.
- 첫 번째 반음절+반음절의 경우는 앞의 반음절을 그 자리의 숫자로 가정한다.
- 두 번째 반음절+반음절의 경우는 뒤의 반음절을 그 자리의 숫자로 가정한다.

3.1.2 후처리 과정을 거친 상태에서의 실험

이번 절에서는 3.1.1 절에서 테스트한 결과를 가지고 불완전한 조합에 대한 숫자음을 재형성한 후, 이를 후처리과정(고립단어인식)을 거쳐서 완전한 결과를 얻었을 때의 인식률을 알아보았다. 표 5를 통해 알 수 있는 것은 후처리를 하지 않았을 때의 인식률인 표 4와 비교해서 많은 인식률의 향상이 있다는 것이다.

표 5. state에 따른 후처리를 거치고 난 인식률 (단위: %)

state 수	자릿수에 따른 인식률	전체 숫자인식률(사연)
8	95.37	84.43
10	96.27	86.23
12	96.59	87.26
13	90.06	79.54

후처리를 거치게 되면 전체적으로 10% 이상의 인식률 향상을 보였다.

불완전한 조합으로 인식결과가 나왔을 때, 불완전한 반음절의 조합들 중 어떠한 것이 실제 인식 결과인지를 알지를 못한다. 그러기에 불완전한 조합에 대해 3.1.1 절에서 정한 규칙은 인식률의 향상에 많은 도움을 주지 못한다. 불완전하게 인식된 결과를 참조하여, 발생 가능한 정상적인 숫자음을 만들어내서 이를 고립단어 인식을 하게 되면, 입력된 음성신호와 유사한 몇 개의 숫자음들에 대해 인식을 하는 것과 같기에 좀 더 정확한 숫자음 인식을 할 수가 있다.

표 6은 후처리과정에서 기존에 제시되었던 방법인, 다음에 인식될 대상을 현재 인식된 결과에 의해 제한을 두는 후처리방법을 사용하였을 때의 인식률을 나타낸 것이다.

표 6. 기존에 제시되었던 후처리과정에 의한 인식률[1] (단위: %)

VQ Level	단위 숫자별 인식률	연속 네 자리 인식률
32	86.1	58.6
64	88.8	65.0
128	88.6	62.8
256	88.7	63.0

표 5와 표 6을 통해 알 수 있는 것은 후처리를 위해 인식된 결과를 토대로 다음 인식대상에 대해 제한하는 것은 후처리를 거치지 않는 것보다 인식률을 향상시킬 수는 있으나, 현재 인식된 대상이 오인식되었을 경우, 오인식을 바르게 보정할 수 없음을 알 수 있다. 하지만 현재 나온 인식대상에 대해 가능한 모든 경우의 수로 숫자음을 재형성하여 그것을 새롭게 고립단어로 인식할 경우는 표5에서 보듯이 후처리를 거치지 않은 것보다 많은 인식률의 향상을 보였다. 이렇게 후처리를 하게 될 경우는 현재의 반음절이 오인식되었을 경우라도, 다음 인식결과중의 반음절을 가져와서 숫자를 형성하기 때문에 오인식되어지는 경우가 줄어들게 되어 인식률이 향상되었다[7].

3.1.3 이중 state의 변화에 따른 실험

앞의 3.1.1과 3.1.2 절에서는 반음절, 반음절+반음절에 대해 동일한 state 수를 사용하였다. 표 4를 보면 state 수가 5일 때 가장 안 좋은 인식률을 보였다. 이는 반음절+반음절의 상태변화를 5 가지로 나타내기엔 부족하기 때문이라고 앞에서 언급하였다. 이와 마찬가지로 단지 반음절로만 이루어진 모델의 state 수를 반음절+반음절로 이루어진 모델에 적합한 state 수와 같게 하는 것도 인식률에 영향을 준다. 표 7을 보면 반음절과 반음절+반음절의 state 수의 차이가 인식률에 어떻게 영향을 주는지를 보여준다.

7) 표 5와 표 6의 인식률 수치를 절대적으로 비교할 수는 없다. 본 논문은 Continuous HMM을 사용하였고, 표 6은 Discrete HMM을 사용하였다. 그러므로, 표 5와 표 6의 인식률 차이가 본 논문에서 제시한 후처리 알고리즘과 기존에 제시된 후처리 알고리즘[1]의 차이라 할 수는 없다. 절대적 수치의 차이보다는 CHMM과 DHMM 간의 차이를 고려하여 상대적인 차이로 인식률의 향상을 보아야 하겠다.

표 7. 이중 state 수를 적용한 인식률 (단위: %)

state 수		자릿수에 따른 인식률	전체 숫자인식률(사연)
8	12	97.46	90.48
10	12	97.46	90.48
11	12	97.14	89.32
10	13	97.30	90.22
11	13	96.91	89.83

표 6과 비교해 보면 인식률이 4~5% 향상되었다. 반음절+반음절의 경우는 표 6에서 state 수가 12일 때 인식률이 가장 좋았기 때문에 이번 실험에서도 state 수를 12로 하여 사용하였고, 반음절에 대한 state 수의 변화로 인식률을 알아보았다. 반음절의 경우 state 수가 8 혹은 10일 때 동일한 인식률을 보임으로써 두 경우가 가장 좋은 인식률을 보였다.

3.2 인식유닛에 따른 인식률 변화

이번 절에서는 인식유닛을 반음절기반으로 하였을 때와 음소기반(Di-phone)으로 하였을 때의 인식률의 차이를 알아보았다. 반음절기반을 인식유닛으로 할 경우 이번 실험에서는 state 수를 이중 state의 경우로 선택하였고, Di-phone의 경우는 state 수를 6으로 하여 테스트하였다.

표 8. 반음절기반과 음소기반에 따른 인식률 (단위: %)

인식유닛	state 수	자릿수에 따른 인식률	전체 숫자인식률(사연)
반음절 기반	10-12	97.46	90.48
음소기반(Di-phone)	6	96.20	85.71

이번 실험에서 사용되어진 각 인식유닛의 state 수는 가장 좋은 인식률을 보여주는 state 수를 선택하여 인식률을 비교하였다. 자릿수에 따른 인식률의 경우 반음절기반이나 음소기반 모두 95% 이상의 좋은 인식률을 보였으나, 전체 숫자인식률의 경우 인식유닛을 Di-phone 기반으로 하였을 경우, 4% 정도의 인식률이 저하되었다. 이러한 저하요인은 Di-phone이 앞, 뒤의 문맥에 따라 인식유닛이 결정되어진다고 하지만 연음현상을 반음절+반음절 만큼 잘 표현하지 못하기에 인식률의 저하를 가져왔다. 연음현상은 두 음절간의 음이 합쳐져서 나게 되는 것인데, 이를 시간상 길이로 본다면 음소보다는 반음절+반음절 유닛[8]이 잘 반영하기 때문이다[7]. 그리고 인식률을 저하시킨 다른 요인으로는 Di-phone으로 음소들을 표현할 때 서로 다른 음소들이지만, 같은 Di-phone이 형성되어 오인식이 되어지는 경우가 생기게 된다. 본 논문에서 Di-phone으로 구현시 '일육'과 '이륙'의 경우 동일한 Di-phone이 형성되어 인식에 오류가 발생하였다.

8) 반음절+반음절 유닛은 앞 음절의 뒷부분과 뒤 음절의 앞부분을 합쳐서 형성하는 유닛이기에 시간의 길이로 보자면 음절과 비슷하다.

4. 최종 결과

앞의 2, 3 절에서는 특징벡터의 파라미터에 따른 인식률과, state 수에 따른 인식률의 차이를 알아보았으며, 인식유닛에 따른 인식률의 결과를 알아보았다. 이번 절에는 앞 절에서 수행한 결과를 다시 정리하고자 한다.

논 논문에서 제시한 반음절 기반의 숫자음 인식기의 경우는 그림 7, 그림 8에서 볼 수 있듯이 후처리 과정을 거치면서 state 수는 이중 state로 반음절은 8, 반음절+반음절은 12차를 사용하거나 반음절은 10, 반음절+반음절은 12차를 사용하는 것이 가장 좋은 인식률을 보이게 된다.

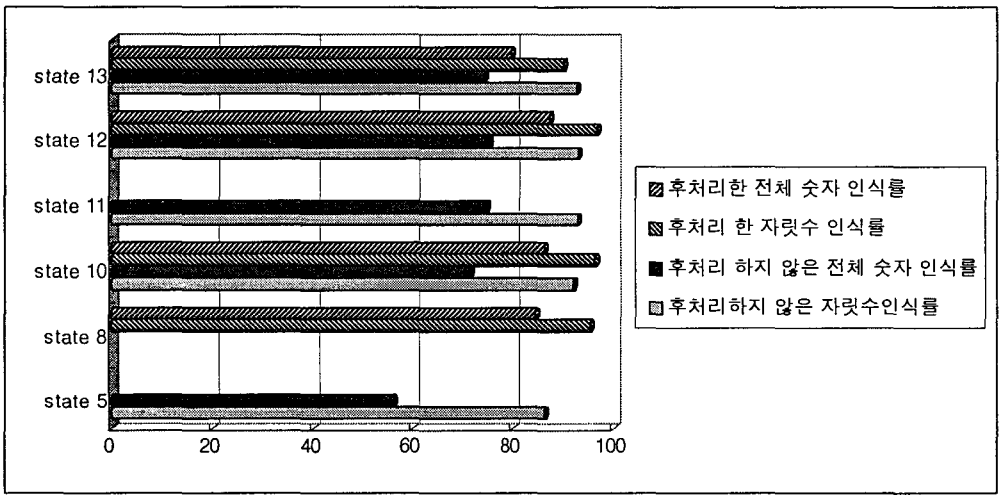

그림 7. 후처리 여부에 따른 인식률 비교

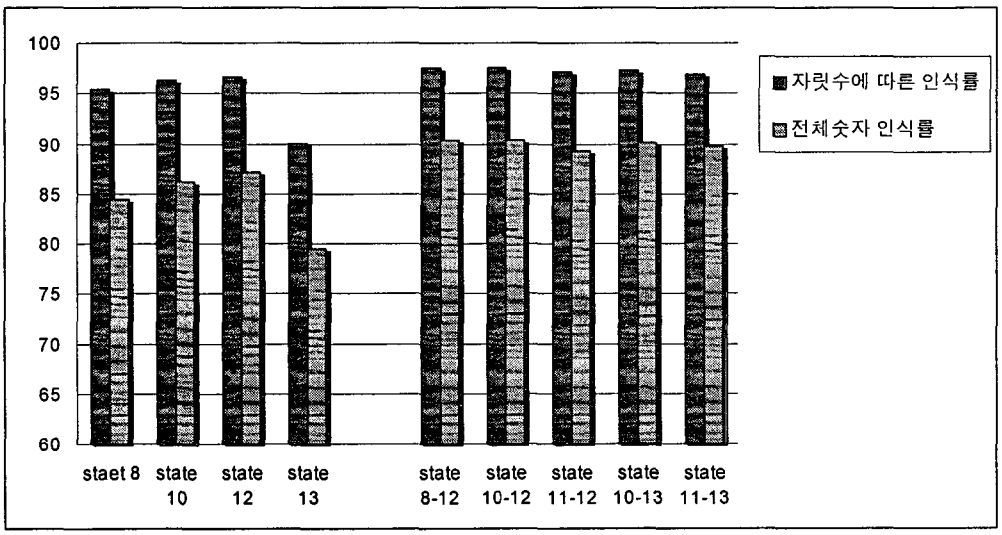

그림 8. 각 state 수에 따른 인식률 비교

또한, 기존에 제시된 반음절기반을 사용하지만, 후처리 과정을 제한적으로 사용한 방법[1]과 본 논문에서 제시한 후처리 기법을 사용하여 인식을 하게 되는 경우의 차이를 비교함으로써 기존의 방식보다 더 좋은 인식률을 가지는 인식기를 구현할 수 있음을 알 수 있다.

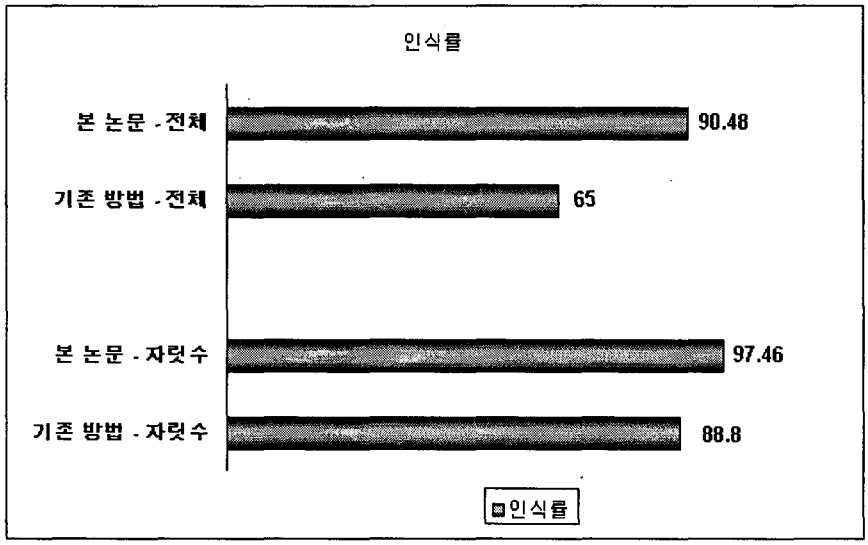

그림 9. 기존에 제시된 후처리[9]와 본 논문에서 제시한 후처리 비교

5. 결 론

본 논문에서는 한국어연결숫자음인식시스템을 구현하기 위한 방법을 제시하고, 그것의 성능을 테스트하였다. 숫자음 인식은 주민등록번호, 전화번호, 신용카드번호 등 현재 생활에서 사용되어지는 것들을 대체하여 사용되어질 수 있다. 하지만 한국어 숫자음의 특성상 정확한 인식은 많은 어려움이 있다. 이를 개선해보고자 인식유닛을 기존에 사용되어져 오던 음소기반이 아닌 반음절기반으로 변경하였고, 지속시간의 변화와 순간적인 오인식에 대한 보정을 위하여 고립단어인식의 후처리를 적용하여 인식률을 향상시켰다. 두 음절이 연속적으로 발음될 때 발생하는 연음현상을 표현하기에는 기존의 음소단위의 표현보다 확대되어진 개념이 필요하였기에 반음절 기반으로 인식유닛을 변경하였고, 반음절로 인식된 결과 중 잘못 인식되어진 결과는 고립단어형식의 새로운 숫자음으로 형성하여 다시금 인식을 거쳤다. 이러한 인식방법은 두 번의 인식과정을 거치게 됨으로 연산시간이 두 배로 필요하게 되는 단점이 있지만, 현재 사용되어지는 시스템의 사양[10]에서는 인식시간의 차이가 느껴지지 않았다.

숫자음인식시스템은 현재 전화를 통해 이루어지고 있는 많은 정보서비스들의 사용상의 불편함을 해소해 줄 수 있는 시스템이다. 숫자패드를 제외하고는 특별한 입력장치가 없는 전화

9) 그림 9에서 기존방법에 대한 인식률은 참고문헌 [1]의 인식률을 표시하였다.
10) 숫자음인식시스템을 구현한 시스템은 Pentium III 500 MHz, 128 MB Memory이다.

상에서 음성을 이용한 숫자의 입력은 많은 이점을 제공할 것이다.

하지만 본 논문에서의 테스트는 16 kHz로 음성을 sampling하여 테스트하였다. 전화망에서는 음성의 sampling rate가 8 kHz로 낮아질 것이며, 전화망채널을 통과함으로써 채널왜곡이 입력신호에 영향을 준다. sampling rate의 변화와 채널에 대한 입력신호의 왜곡에 대한 보정이 전화망상에서의 인식률을 결정할 것이며, 이를 어떻게 보완하여 인식률을 향상시킬지는 앞으로 실용적인 전화망 숫자음인식시스템 구현에 해결해야할 과제이다[8][9].

또한, 현재 구현된 숫자음인식시스템은 인식률을 저하시키는 연음현상에 대한 부분을 인식유닛의 변경으로 인식률을 향상시켰지만, 주변잡음에 대해서는 아직 오인식되는 경우가 있다. 주변잡음에 대한 처리가 실용화되어 사용하게 될 숫자음 인식 시스템에 대해 남아있는 중요한 과제이다.

본 논문에서는 사연숫자음에 대해서만 인식률테스트를 하였다. 이를 좀더 확장하여 여섯자리, 일곱 자리, 여덟 자리 등으로 확장하여 사용하면 좀 더 다양한 숫자음인식시스템에 적용할 수 있다.

참 고 문 헌

[1] 윤재선, 홍광석. 1998. "반음절 단위 HMM을 이용한 연속 숫자 음성인식." *한국음향학회지*, Vol. 17 No.5 , 73-78.

[2] Davis, S. B. & P. Mermelstein. 1980. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoustics, Speech, Signal Proc*, ASSP-28 (4), 357-366.

[3] Rabiner, L. R., B. H. Juang., S. E. Levinson., & M. M. Sondhi. 1985. "Recognition of isolated digits using hidden Markov models with continuous mixture densities," *AT&T Technical Journal*, vol. 64, 1211-1234.

[4] Junqua, J. C. & J. P. Haton. 1996. *Robustness in Automatic Speech Recognition*, Kluwer Academic Publishers, Boston.

[5] Lee, K. F. 1989. *Automatic Speech Recognition - The Development of the SPHINX System*, Kluwer Academic Publishers, Boston.

[6] Young, Steve. 1996. "Large Vocabulary Continuous Speech Recognition: a Review." *Technical report, Cambridge University Engineering Department*, Cambridge, UK.

[7] Wu, S.-L., B. E. D. Kingsbury., N. Morgan. & S. Greenberg, 1998. "Performance improvements through combining phone- and syllable-scale information in automatic speech recognition." ICSLP.

[8] Moreno, P. & R. Stern. 1994. "Sources of Degradation of Speech Recognition in Telephone Environments." *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 109-112.

[9] Falavigna, D. & R. Gretter. 1997. "Evaluation of Digit Recognition Over the Telephone Network." *In Proceedings of ESCA Workshop on Robust Speech Recognition for Unknown Communication Channels*. Pont-a`-Mousson. France.

접수일자: 2001. 10. 27.
게재결정: 2001. 12. 5.

▲ 정재부
강원도 춘천시 강원대학교 정보통신연구소 302호 (우: 200-701)
강원대학교 전자공학과
Tel: +82-33-250-6322
E-mail: miru@dsplab.kangwon.ac.kr

▲ 정 훈
강원도 춘천시 강원대학교 정보통신연구소 302호 (우: 200-701)
강원대학교 전자공학과
Tel: +82-33-250-6322
E-mail: hchung@mirae.kangwon.ac.kr

▲ 정익주
강원도 춘천시 강원대학교 정보통신연구소 302호 (우: 200-701)
강원대학교 전자공학과
Tel: +82-33-250-6322
E-mail: ijchung@kangwon.ac.kr