한국어 연결단어의 이음소 인식과 어절 형성에 관한 연구*
A Study on the Diphone Recognition of Korean Connected Words and Eojeol Reconstruction

김 경 선*, 정 홍**
(Kyeong Sun Kim*, Hong Jeong**)

*이 논문은 1993년도 한국학술진흥재단의 공모과제 연구비와 정보통신연구소의 개발과제 연구비에 의하여 연구되었음.

요 약

본 논문에서는 시간지연신경망을 이용한 한국어 무제한 어휘 연결단어 인식 시스템에 대해 기술하였다. 인식단위로는 인접한 두 음소의 천이과정을 포함하는 이음소(diphone)를 사용하였으며 그 갯수는 329개이다. 한국어 연결단어 인식과정은 음성신호의 특징 추출 과정, 이음소 인식과정과 후처리 과정의 세 단계로 구분된다. 특징 추출 단계에서는 입력 음성의 이음소 구간을 분리하여 16차의 필터뱅크(filter-bank) 계수를 구한다. 이음소 인식은 3단계의 계층적 구조로 이루어졌으며 총 30개의 시간지연신경망을 이용해 이음소를 인식한다. 특히, 사용된 시간지연신경망은 인식률을 높이기 위하여 기존의 시간지연신경망 구조를 변경하였다. 후처리 단계는 음소 천이확률과 음소 혼돈확률을 이용한 이음소 오인식 수정과정과 인식된 이음소를 결합하여 어절을 형성하는 과정으로 이루어진다.

ABSTRACT

This thesis described an unlimited vocabulary connected speech recognition system using Time Delay Neural Network(TDNN). The recognition unit is the diphone unit which includes the transition section of two phonemes, and the number of diphone unit is 329. The recognition processing of korean connected speech is composed by three parts : the feature extraction section of the input speech signal, the diphone recognition processing and post-processing. In the feature extraction section, the extraction of diphone interval in input speech signal is carried and then the feature vectors of 16th filter-bank coefficients are calculated for each frame in the diphone interval. The diphone recognition processing is comprised by the three stage hierachical structure and is carried using 30 Time Delay Neural Networks. Particularly, the structure of TDNN is changed so as to increase the recognition rate. The post-processing section, mis-recognized diphone strings are corrected using the probability of phoneme transition and the probability of phoneme confusion and then the eojeols(Korean word or phrase) are formed by combining the recognized diphones.

*포항시 포항공과대학 전자전기공학과
Department of Electrical Engineering, POSTECH
**포항시 포항공과대학 전자전기공학과
Department of Electrical Engineering, POSTECH
접수일자 : 1995년 5월 6일

I. 서 론

1980년대 이후 등장한 신경망은 인간의 두뇌의 생물학적 신경 계통에 근거한 간단하고 많은 처리요소를 병렬로 연결하여 이루어진 것으로, 학습을 통해 입력패턴에 내재하는 정보를 처리하는데 용이하여 현재 신경망을 이용하려는 연구가 여러 분야에서 다양하게 시도되고 있다[5, 12]. 특히 음성인식에 있어서 신경망의 사용은 여러 가지 장점이 있는데 신경망은 인간의 신경계통을 모방한 것으로 인간이 음성을 인식하는 방법과 근접하며, 하드웨어로 구현된다면 병렬 처리가 가능하여 많은 계산을 빨리 할 수 있다. 또한, 작은 에러나 음성신호의 잡음 등에 의한 영향에 비교적 잘 견디어 낼 수 있다. 그리고 신경망의 구조를 변화시켜 원하는 특성을 잘 감지할 수 있도록 구현하기가 용이하다. Waibel[11, 16]에 의해 제안된 시간지연 신경망은 음성신호의 특성인 시간굴곡 현상과 시간지연 현상을 용이하게 해결할 수 있도록 설계된 것으로 신경망의 내부구조를 적절히 바꾸어 구현한 대표적인 예라 할 수 있다. 또한, "보다 넓은 기능의 추가를 위하여 확장이나 모듈별 구성이 용이하다"는 점을 [8, 17, 18] 들 수 있다.

본 논문은 대화체 음성인식을 궁극적인 목표로 하여 이것의 첫 단계로 연속음성 인식과 대용량 음성인식을 가능하게 하는 방법과 인식된 결과를 언어처리 시스템으로 넘겨주는 중간 단계로서 어절을 만드는 방법에 대해 살펴본다. 본 논문에서는 연속음성인식과 대용량 음성인식을 가능하게 할 뿐만 아니라 음성인식기 성능을 향상시키기 위해 diphone이란 인식단위를 채택했다. 많은 시스템에서는 연속음성인식에 음소 단위를 사용했지만 그 안에 내포하고 있는 정보가 부족해 인식에 어려움이 많다. Diphone은 두 음소의 결합 형태로 생각할 수 있으며 가지고 있는 정보도 각각의 음소 정보와 그 사이의 음소 천이 정보가 있어 음소보다 인식에 유리하다. 한편, 기존의 시간지연신경망의 구조를 변경하여 인식률의 향상을 꾀하였다. 이렇게 인식된 결과를 언어처리 시스템으로 넘겨주는 중간 단계로서 음성 인식 결과의 오인식 수정과 어절 형성 과정을 수행한다. 오인식 수정은 diphone 인식기의 특성을 모방한 음소 혼돈 확률과 한글 발음의 음운 규칙을 모델링한 음소 천이확률을 이용해 확률적으로 접근했다. 오인식 수정 과정을 거친 음소열을 결합하여 발음 가능성을 조사한 후 최적의 어절을 찾아 출력으로 내보낸다.

2장에서는 인식 시스템의 전체 구조 및 인식단위인 이음소(diphone)의 특성과 분리 과정을 설명하고 3장에서는 시간지연신경망의 특성과 개선된 신경망의 구조 그리고 이음소 인식기에 대해 설명한다. 4장에서는 인식 실험 및 그 결과에 대해 설명하고, 5장에서는 음성인식 후처리 과정에 대하여 설명한다. 마지막으로 6장에서는 이음소 인식기의 특성과 응용분야, 앞으로의 연구 방향에 대해 설명한다.

II. Diphone을 이용한 무제한 어휘 인식 시스템

무제한 어휘 인식 시스템은 한국어의 변형된 329개의 diphone 인식을 위한 것으로 30개의 신경망으로 구성되어 있다. 이러한 구조는 여러 가지 다양한 실험의 결과를 비교 분석하여 인식률뿐아니라 일반적인 워크스테이션상에서 실시간 처리 능력도 고려하여 선택하였다. 이번 장에서는 diphone 인식 신경망의 전체 구조 및 구조적 특성과 인식단위의 특징에 대하여 설명한다.

II.1 무제한 어휘 음성 인식시스템의 전체구조

본 논문에서 구현한 전체시스템에 대한 구성도는 그림 1과 같다.

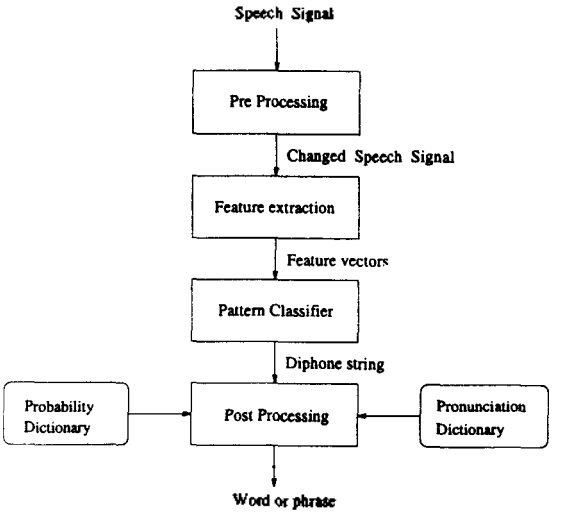


그림. 1. 무제한 어휘 음성 인식기 전체시스템 구성도.
Fig. 1. Overall structure of unlimited vocabulary speech recognizer.

음성신호 전처리에서는 마이크폰 통해 들어온 음성신호를 양자화(quantization)하고, 음성의 질을 높이기 위해 고대역 필터를 거친다. 특징점 추출(feature extraction) 과정에서는 음성인식기의 입력이 될 16차의 filter-bank coefficient를 생성한다. 패턴 분류(Pattern Classifier) 단계는 시간지연신경망을 이용하여 한국어 diphone 열을 구한다. 후처리 과정은 확률 사전과 발음 사전을 이용하여 인식된 diphone 열이 음운규칙과 언어학적인 문법에 맞는 지 검토 수정하여 어절을 출력한다.

Ⅱ.2 인식단위

가장 기본적인 음성인식 시스템은 기본적으로 단어 단위로 인식할 수 있도록 되어 있다. 이것은 음소나 음절등의 서브워드 단위의 음성인식이 인식시간이나 인식률 등의 문제가 있어 아직 실용화가 어려웠던 반면, 단어 단위의 음성인식은 비교적 계산량이 적어 일반적인 컴퓨터상에서 실시간 처리가 가능하였다. 또한 단어 자체로 인식을 할 경우 음소나 음절의 서브워드 보다 많은 정보를 가지고 있어 높은 인식률을 얻을 수 있다. 그러나, 이와 같은 단어 단위 인식 방법은 인식 어휘수가 제한되어 한정된 단어를 사용하는 특수한 응용 분야에서만 이용할 수 있다는 단점이 있다. 이렇게 해서 등장한 것이 음절(syllable), 반음절(demisyllable), triphone, diphone 등의 서브워드 단위이다. 이들은 음소와 마찬가지로 대어휘 음성인식에 적합하고 연속음성인식이 가능하지만, 단어인식의 경우보다는 현저하게 인식률이 떨어진다.

표 1은 각 인식 단위의 특성을 분류한 것이다.

학습성(trainability)이란 인식 단위가 학습이 충분할 정도로 빈번히 발생하는가의 여부를 나타내며, 일괄성(consistency)은 음운동화현상에 관계없이 일괄된 독특한 성질을 내포할 수 있는가의 여부를 나타내며, 분별성(discrimination)은 다른 클래스와 현저히 달라 쉽게 분류할 수 있는가의 여부를 나타내며, 확장성(expansibility)은 인식 단위를 결합하여 올바른 문장을 얻어낼 수 있는지의 여부를 나타낸다. 그리고, 클래스 수는 인식 단위의 수를 의미한다. 단어 단위는 학습성과 클래스 수의 취약점 때문에 무제한 어휘 연속음성 인식에는 사용할 수 없으며, 음소 단위는 음운동화현상때문에 일괄성이 다른 인식단위에 비해 현저하게 떨어진다. 음절이나 triphone, diphone 단위는 서로 비슷한 성능을 보이지만 인식 단위 수에서 diphone이 가장 적기때문에 신경망을 이용할 경우 음절이나 triphone을 인식단위로 하였을 경우보다 학습 시간과 인식률 면에서 유리하리라 생각된다. 특히 본 논문에서 사용한 diphone 단위는 발음의 길이가 100~150 msec로 거의 비슷한 크기이다. 이것은 사용한 diphone이 모든 경우에 중성을 포함하기 때문이다. 인식단위의 시간적인 차이가 거의 없다는 이러한 사실은 일괄성(consistency) 문제와 시간 굴곡 문제를 해결하는 데 잇점이 있고 diphone의 위치를 찾을 수 있는 정보로 사용되었다.

한국어 diphone의 갯수는 이론상으로 음소 48개, 초성과 중성의 결합 399개, 중성과 종성의 결합 168개, 종성과 초성의 결합 152개, 모두 767개이다. 하지만 실제 발음에 쓰이고 발음시간을 고려해본다면 모음 음소 17개, 초성과 중성의 결합 220개, 중성과 종성의 결합 92개, 총 329개로 줄일 수 있다. 즉, 모음별로 이음소를 분류하여 /애, 에/, /외, 왜, 웨/의 모음 그룹을 /애/, /외/의 대표 모음 그룹에 포함시켜 이음소 갯수를 줄였다. 또, diphone 중 종성과 초성의 결합 꼴은 diphone 목록에서 제외했다. 이것은 종성과 초성의 결합 꼴은 발음시간이 짧아 입력 구간을 선택할 때, 중성 부분이 반드시 들어가게 되므로 다른 diphone 그룹과 혼동을 일으켜 인식기의 인식률을

표 1. 무제한 어휘 연속음성인식을 위한 인식단위별 특징.
Table 1. The recognition units properties for unlimt vocabulary continuous speech recognizer.

	Phoneme	Syllable	Word	Triphone	diphone
Trainability	Good	Average	Bad	Average	Average
Consistency	Bad	Average	Good	Average	Average
Discrimination	Bad	Average	Average	Good	Good
Expansibility	Bad	Average	Good	Average	Average
Class Number	Good	Bad	Bad	Bad	Average

낮추는 요인으로 작용되며 나중에 음절 분라 알고리듬을 이용하여 보완할 수 있기 때문이다.

Ⅱ.3 음성 신호의 특징 추출

음성인식을 위한 학습패턴생성 및 인식을 위한 패턴 생성은 음성 신호 전처리(pre-processing), 특징 파라미터 추출(parameteric representation) 그리고 정규화(normalization) 과정을 거친다.

양자화(quantization)된 음성신호는 $1-0.95z^{-1}$의 형태를 가지고 고대역 필터를 거치는데, 이것은 음성신호를 양자화하는 과정 중 저대역 필터를 거칠 때 손실되는 고대역 부분을 보상해준다. 그 다음 주파수 영역에서 분석하고, 이를 바탕으로 인식에 용이한 형태로 변환한다.

본 시스템은 음성신호의 주파수 영역을 분석하는 방법으로서 Short-Time Fourier Transform(STFT)을 사용한다. 음성신호는 시간에 따라 변화하며 그 변화도 불규칙적이다. 그리고 그 음성신호를 발생하는 모델도 시간에 따라 변화하게 된다. 그 결과 음성신호는 어떤 짧은 시간동안 준주기성(quasi-periodic)을 가지며 그 준주기성은 전체적으로 변화한다. 하지만 대부분의 음성신호분석에서 가정하는 것은 "음성신호의 특성이 짧은 시간에 급격히 변화하지 않고 서서히 변한다"는 것이다. STFT는 이와같은 가정하에 짧은 구간의 음성신호에 대한 Fourier Transform을 정의한 것이다. 실제 STFT의 계산은 Hamming Window를 취한 음성구간에 대해 FFT 이용해 주파수 성분을 계산한다.

일반적인 음성인식 시스템은 주파수 영역에서 분석되어진 결과를 바탕으로 실제 음성신호의 정보를 잘 나타내고 필요없는 부분을 제거하기 위하여 음성신호의 특성 파라미터를 추출하는데, 이것은 실제 음성인식시스템을 구현하는데 매우 중요한 문제이다. 그리고 여러가지 실험에 의해서 다양한 결과가 나타나고 있지만[2, 9], 절대적인 기준은 정하기가 힘들고 실제 구현되는 시스템의 특성에 맞추어져야 한다.

본 시스템의 특징 추출(feature extraction)은 STFT을 하여 얻어진 주파수 성분으로 부터 포만트나 포만트 천이 과정 등의 음성인식을 위한 중요한 정보를 추출하는 것으로서 16차의 필터 뱅크 계수(filter-bank coefficient)를[3] 이용한다. 이것은 FFT된 64개의 성분을 mel-scale된 16개 구간의 에너지로 나타낸 것으로 각 대역 필터의 주파수 영역이 표 1에 나타나 있다.

이 때 boundary effect를 줄이기 위하여 16개의 구간을 각 구간의 반 만큼 이동하여 전체적으로 32개의 성분을 만들어 이웃하는 성분과 평균하여 최종적으로 16개의 성분을 구한다.

실제 사용된 입력 벡터는 16 msec 크기의 음성 구간에서 필터 뱅크 계수를 구하고 8 msec씩 이동하여 총 171 msec 동안 32개의 성분을 구한다. 이러한 32개의 입력 벡터들을 2개씩 취하여 평균을 구해 최종적으로 16개의 입력벡터를 얻는다. 따라서, 각 diphone을 위한 입력패턴은 16×16의 필터 뱅크 계수로 나타낼 수 있다.

추출된 파라미터를 신경망의 입력패턴으로 사용하기 위하여서는 입력패턴의 값을 −1과 1사이의 값이 되도록 정규화(Normalization)해야 한다. 아울러, 사

표 2. n = 16 일때의 Mel-scale된 주파수 밴드(8KHz sampling)
Table 2. Mel-scaled frequency band when n = 16.

FFT Points	Frequency 성분(Hz)	FFT Points	Frequency 성분(Hz)
2−3	150−260	22−25	1398−1628
4−5	261−381	26−29	1629−1880
6−7	382−514	30−33	1881−2158
8−9	515−660	34−35	2159−2462
10−12	661−819	39−43	2463−2795
13−15	820−994	44−49	2796−3160
16−18	995−1186	50−56	3161−3560
19−21	1187−1397	57−64	3561−4000

간적으로 이웃하는 성분의 크기와 상대적인 변화를 유지하면서, 입력 패턴의 각 값들이 −1과 +1사이가 되도록 정규화되어야 한다. 임의의 입력벡터의 한 원소를 I_{ij}라 하면, 다음과 같이 입력벡터의 크기를 구하여 모든 성분을 정규화한다.

$$E(I_{ij}) = \frac{1}{16}\sum_{i=1}^{16}\frac{1}{16}\sum_{j=1}^{16} I_{ij}, \quad (1)$$

$$\| I_{ij} \| = \sum_{i=1}^{16}\sum_{j=1}^{16} | I_{ij} - E(I_{ij}) |^2, \quad (2)$$

$$I_{ij} = \frac{I_{ij}}{\| I_{ij} \|}, \quad i=1, \ldots, 16 \quad j=1, \ldots, 16, \quad (3)$$

Ⅱ.4 Diphone 인식기의 계층적 구조

Diphone 인식기만의 전체 구조는 그림 1에 나타내었다.

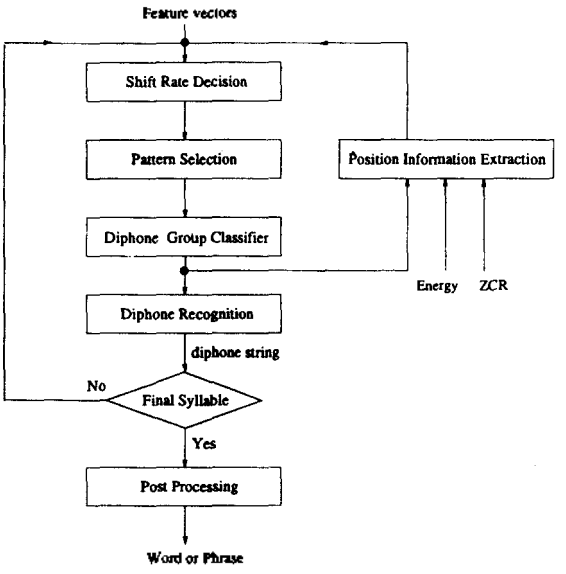


그림. 2. Diphone 인식 시스템의 전체 구조
Fig. 2. The sturcture of diphone recognizer.

Diphone 인식기의 입력으로서는 단구간에너지(short-time energy)와 영교차율(zero-crossing rate)과 필터뱅크 계수이고 출력은 diphone 열이 된다. 이 과정은 음성구간의 이동거리를 계산하는 과정과 세 단계의 TDNN으로 구성 된다.

음성구간의 이동거리를 계산하는 과정은 점진적 이동 음성 인식의 단점인 인식시간과다 문제를 해결하기 위해 어절내에서 diphone의 위치를 찾아내 그 부분만 신경망의 입력으로 하여 필요없는 계산시간을 줄이는 것을 목표로 둔다. Diphone 위치는 첫단계 신경망의 출력, 구간에너지 그리고 영교차율을 이용하여 알아낸다.

실제 diphone을 인식하는 단계에서는 diphone의 갯수가 많기 때문에 하나의 신경망으로는 시스템을 구현하기는 어렵다. 따라서 3 단계의 계층적 구조[13]를 갖는 신경망 구조를 도입하였다. 이로써 학습의 분산, 인식 시간의 단축 등의 잇점을 얻었지만 앞 단계에서의 잘못은 그 다음 단계로 그대로 전파되어 오인식 수정에 어려움이 있게된다. 이 문제를 해결하기 위해 diphone 3개를 가지고 음절로 만들어 오인식 수정을 시도하는 방법을 사용하였다. 첫 번째 신경망은 위치 신경망(position net)으로서 diphone의 음절내의 위치를 판가름해주는 역활을 하고, 두 번째 신경망은 그룹 신경망(group net)으로서 해당 음성구간의 안정된 특성이 무엇인지 즉 어떠한 모음 성분을 포함하고 있는지 알아내는 역활을 하며, 마지막 단계의 신경망은 천이 신경망(transition net)으로서 앞의 두 단계의 결과를 이용해서 해당 음성구간 내에 어떠한 diphone 성분이 들었는지 알아낸다. 이러한 diphone 인식을 위한 각 계층의 전체 신경망 이름을 그림 1에 나타내었다.

그룹1 신경망과 그룹2 신경망의 출력은 17개의 모음 인덱스이고 그룹3 신경망의 출력은 9개의 단모음 인덱스이다. 전체 신경망의 갯수는 위치 신경망과 3개의 그룹 신경망 그리고, 26개의 천이 신경망 이렇게 30개가 된다. 그리고, 각각의 신경망 구조는 그림 1과 같다.

Ⅱ.5 Diphone 분리과정

입력된 임의의 신호로부터 음성신호를 분리해 내는 작업은 연속음성인식의 시간적인 문제를 해결하기 위해서 반드시 필요한 부분이며, 때에 따라서는 음성인식에 중요한 요소로 작용한다. 그리고 음성데이타 구축에 있어서도 묵음(silence)를 미리 제거하여 저장할 수 있어 효율적인 작업을 할 수 있게 한다. 이번 장에서는 인식의 대상인 이음소(diphone)를 분리하는 과정에 대해 설명한다. 이는 우선 음성부분을 분리하고 다시 음절로 분리한 후 음절안에서 이음소를 분리하는 과정을 거친다.

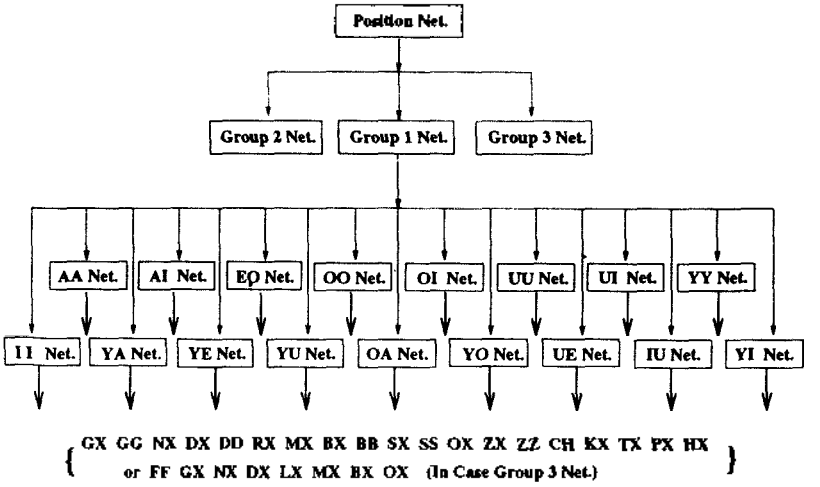


그림. 3. Diphone 인식을 위한 계층별 신경망 구조.
Fig. 3. The hierachical network structure for recognizing diphone.

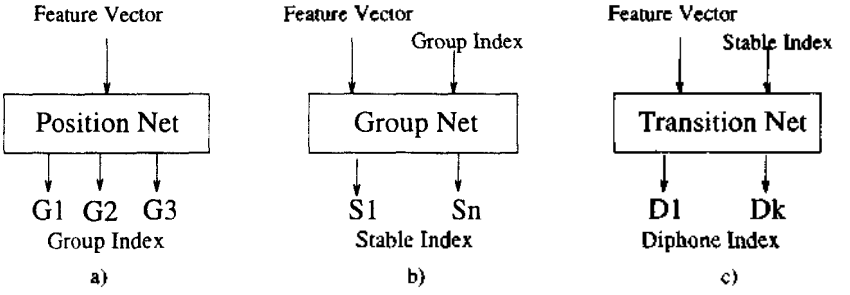


그림. 4. 각단계의 신경망 구조.
(a) Position Net. (b) Group Net.
(c) Transition Net.
Fig. 4. The network structure of each step.
(a) Position Net. (b) Group Net.
(c) Transition Net.

본 연구에서는 음성신호 분리과정으로 Explicit형[10] 끝점검출기를 사용하였는데, 이것은 인식 알고리듬에서 사용되는 시간지연 신경회로망이 음성의 정확한 끝점위치에 상관없이 인식을 할 수 있다는 특성을 가지고 있기 때문이다. 구현된 끝점검출기는 크게 adaptive level equalizer와 에너지 펄스 검출부로 구성되어 있다.

끝점의 검출은 주로 short-time energy와 영교차율(zero crossing rate)를 이용하여 이루어지며, 이들 값은 다음과 같이 정의된다.

$$E(n)=\sum_{n-N+1}^{n} x(k)^2, \tag{4}$$

$$Z(n)=\sum_{n-N+1}^{n} |x(k)-x(k-1)|. \tag{5}$$

여기서 $x(n)$은 양자화된 입력신호의 값을 나타내며, N은 short-time energy의 계산을 위한 윈도우 길이를 나타낸다.

위에서 정의된 에너지와 영교차율은 주변환경이나 녹음상태에 따라서 항상 가변적이다. 따라서 이들 값

늘은 배경잡음(background noise leve)에 의해서 조정되어져야 하는데 이것을 adaptive level equalization이라 하며, 이것은 다음과 같은 식에 의해 행하여진다.

$$E(n) = 20\log_{10}(E(n)) - Q_e \quad (6)$$

$$Z(n) = 20\log_{10}(Z(n)) - Q_z \quad (7)$$

여기서 Q_e와 Q_z은 배경잡음의 최대치를 나타낼 수 있도록 결정된 값이다.

Adaptive Level Equalize된 에너지와 영교차율은 잡음이 들어오면 0을 중심으로 변화하는 값이 되고, 음성신호가 들어오면 큰 값을 갖게 된다. 따라서 미리 결정된 문턱값(threshold)으로부터 음성신호에 해당하는 에너지 펄스를 검출하고, 검출된 에너지 펄스에 대하여 음성신호로 적합한지 검사하여, 적합한 경우에 대하여 음성신호의 시작점과 끝점을 찾는다. 그 다음 에너지로부터 찾아진 시작점에서 영교차율을 검사하여 정해진 문턱값을 넘어서면 문턱값보다 낮아지는 점을 찾아새로운 시작점으로 간주한다. 이것은 무성음이 있는 경우 시작점을 보다 정확히 찾기 위함이다.

끝점 검출과정에서 사용되는 문턱값의 결정은 실험적으로 미리 구하여지는데, 이중 찾아진 에너지 펄스에 대하여 음성신호로 적합한지를 결정하는데 에너지 펄스의 최대값에 대한 문턱값과 에너지 펄스구간의 문턱값은 다른 문턱값과는 달리 배경잡음의 크기에 따라 달라져야 한다. 이것은 처음 adaptive level equlaize 과정에서 배경잡음이 지나치게 심한 경우 음성신호의 에너지 펄스가 상대적으로 작아져 음성신호의 결정되지 않는 가능성이 발생하기 때문이다.

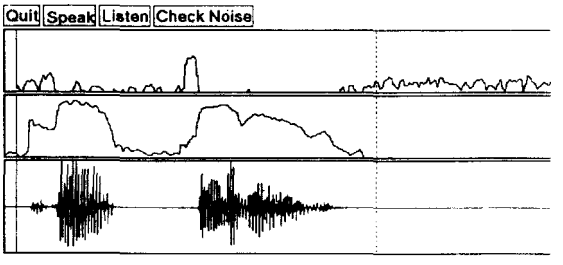


그림. 5. 끝점 검출 과정 예(/각질/).
Fig. 5. The end point detecting procedure.

다음 그림 1은 간단한 어절에 대해 끝점을 검출한 결과를 덤프한 것이다.

맨 위의 윈도우는 영교차율을, 가운데 윈도우는 음성의 단구간 에너지를, 마지막 윈도우는 음성의 파형을 각각 그린 것이다. 이 알고리듬을 통하여 그림 1에서 보는 바와 같이 전체 입력 중에서 음성 부분이 검출되었으며, 검출된 음성 부분은 diphone 인식기의 입력으로 들어가 diphone 열로 바뀐다. 최종적으로, diphone 열은 어절 형성 시스템을 통해 어절 혹은 언절*로 된다. 이렇게 하여, 분리된 음성 부분은 하나의 어절 혹은 언절로 바뀐다.

다음은 음절의 위치와 diphone 인식기의 첫단계인 Position Network의 결과를 사용하여 음절내에서의 diphone의 위치를 알아내는 과정이다. 그림 1은 음성입력을 받아 영교차율(zero crossing rate)과 단구간 에너지(short time energy)을 음성 파형과 함께 그린 것이다. 그림 1은 영교차율과 단구간에너지를 이용하여 대략적인 음절 분리를 하여 나타낸 것이고,

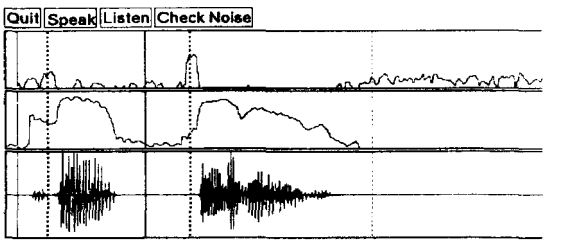


그림. 6. /각질/이란 음성의 음절 분리.
Fig. 6. Syllable segmentation.

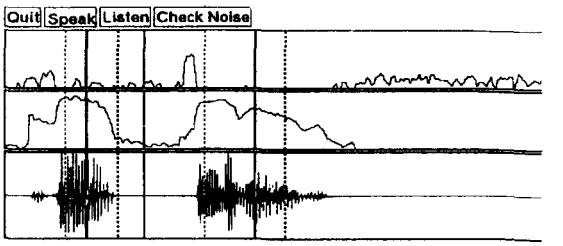


그림. 7. /각질/이란 음성의 diphone 분리.
Fig. 7 Diphone segmentation.

*발음의 편의상 나타나는 한국어 어절의 결합 혹은 어절의 분리 형태

그림 1은 위치 신경망의 출력을 이용하여 diphone의 음절 내에서의 위치를 표시한 것이다.

이러한 방법의 장점은 각 신경망의 입력 세그먼트 갯수가 줄어들어 인식시간에 큰 이득을 하고 아울러 학습한 패턴들과 비슷한 위치의 입력들을 취하는 것이므로 인식률을 향상에 도움을 준다.

Ⅲ. Time-Delay Neural Network

이번 장에서는 diphone 인식의 방법으로 채택된 시간지연 신경망의 특징과 새롭게 제안된 시간지연 신경망을 소개한다.

Ⅲ.1 시간지연신경망의 특징

음성인식에 사용되는 알고리듬 중 현재 가장 뛰어난 성능을 보이는 것은 HMM(Hidden Markov Model)이다. HMM의 특징은 Viterbi 알고리듬이란 디코딩 방법과 Bawm-Welch 알고리듬이란 지도학습을 들 수 있다. 그러나, HMM은 음성신호모델을 정확히 구현할 수 없어 유사한 입력들을 구분하는데 다소의 어려움이 있다는 단점이 있다. 아울러 HMM은 하드웨어 구현이 어렵고 계산량이 많으며 메모리를 많이 요구하는 단점이 있다.

시간지연신경망은 일반적인 다층신경망(Multi-Layer Perceptron)에 지연요소를 첨가하여 음성의 동적인 특성을 위치에 관계없이 감지하도록 구성된 시스템이다. 시간지연신경망과 다른 알고리듬의 특성 [11, 16]을 살펴보면 표 1와 같다.

표 3. 인식 알고리듬의 비교.
Table 3. Comparison of recognition algorithm.

	TDNN	HMM	DTW
Time Shift Invariance	Good	Bad	Bad
Time Warping	Average	Average	Average
Learning Capability	Good	Average	Average

TDNN을 이용한 방법이 time-shift invariance과 학습의 잠재력 면에서 가장 좋다.

한편, TDNN에서 하나의 처리요소는 현재의 벡터 f(t)와 함께 시간적으로 지연된 f(t-1), f(t-2),...,f(t-d)를 동시에 입력으로 받아들이고, 이들에 각각 연결강도(weight)를 곱하여 합한 것을 활성화 함수를 통하여 출력한다. 따라서 시간축상에서 d+1개의 이웃한 입력들의 관계를 용이하게 처리할 수 있어 음성신호의 특징인 시간굴곡현상(time-warping)을 감지할 수 있다. 이때 사용되는 활성화 함수는 대부분의 경우 일반적인 시그모이드(sigmoid) 함수이지만, TDNN의 구조적인 변화와 학습시 수렴성의 향상을 위하여 다양한 형태의 함수가 제안되어 사용된다.[6, 7, 15]

Ⅲ.2 새로운 구조의 시간지연 신경망

음소와는 달리 diphone은 인식단위 갯수가 많으므로 인식률을 높이기 위해서는 음성신호로부터 많은 정보를 끄집어내야하고 그 정보를 적절히 보관할 수 있어야 한다. Diphone은 음소보다 긴 시간동안 음성의 특성을 감지할 수 있다. 하지만, 300개가 넘는 diphone을 인식하기 위해서는 이것으론 부족하다. 그래서, 좀더 많은 정보를 끄집어내고 그것을 보관하기 위해서 시간지연 신경망의 구조를 약간 바꾸었다. 바뀐 신경망의 구조는 그림 1에 나타나 있다.

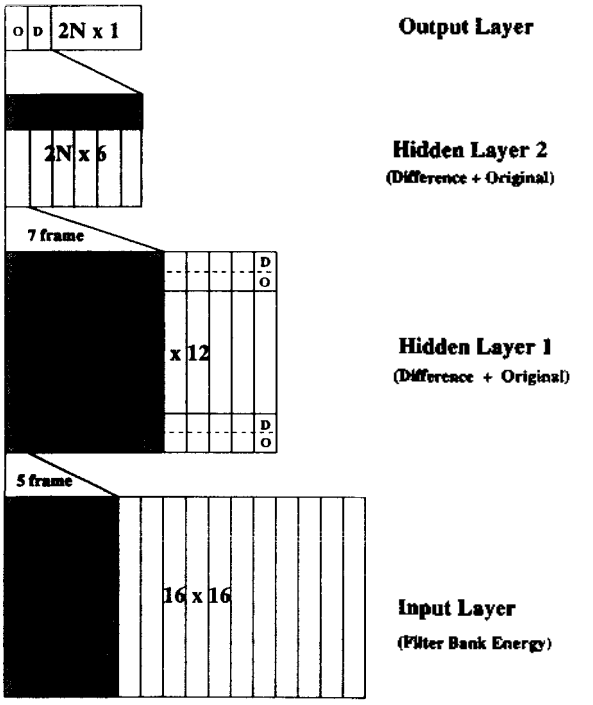


그림. 8. 새로운 구조의 시간지연신경망의 구조.
Fig. 8. New TDNN structure.

본 시스템은 이러한 신경망 30개를 이용하여 329개의 diphone을 인식한다.

일반적으로 패턴인식을 위한 신경망회로과는 달리 효과적인 음성인식을 위하여서는 사용된 신경망의 입력은 주파수 축으로 16개의 성분 그리고 시간 축으로 16개의 성분을 가진다. 인식단위로 선택된 변형된 diphone의 특성은 음소의 특성과는 달리 제1포만트와 제2포만트의 천이과정에 의해 주로 결정되며, 주파수 성분이 안정된 부분이 반드시 존재한다는데 있다. 따라서, 그림 1에서 보듯이 필터뱅크 계수의 차이값을 시간지연신경망의 입력에 직접 인가하여 음성의 동적인 특성을 잘 감지하게 하였다.

기본적인 구조는 일반적인 시간지연신경망과 같으나, 각 층의 노드와 노드를 연결하는 방법은 다르다. 시간적으로 같은 위치에 있는 노드들의 집합(즉, 앞에서 입력벡터 f(t)에 해당)을 프레임이라고 할 때, 은닉층(hidden layer) 1의 노드들은 시간적으로 이웃하는 5개의 프레임으로부터 입력을 받는다. 그런데, 은닉층 1의 노드들은 원래의 입력값과 이웃하는 프레임사이의 차이값 각각에 대해서 가중치(weight)를 계산하여 저장한다. 즉, 가중치(weight)는 필터뱅크 계수의 분포에서 특징을 찾아내는 가중치(weight)와 필터뱅크 계수의 차이에서 특징을 찾아내는 가중치(weight)로 나누어진다. 이렇게 함으로서 은닉층 1은 음성의 동적인 특징인 시간 굴곡을 잘 반영하게 된다.

그리고, 은닉층 2에서는 은닉층 1의 7개 프레임으로부터 은닉층 1에서와 같은 맥락으로 원래의 입력과 이웃한 입력의 차이를 받아 좀더 시간적으로 긴 특징을 감지한다. 최종적으로 출력층은 은닉층 2의 시간에 대한 6개의 노드들의 출력을 제곱하여 더한 것으로 은닉층 2에서의 한 프레임의 노드수는 출력노드수와 같다. 그림 1의 TDNN의 전체적인 구조는 은닉층 1과 입력층, 은닉층 1과 은닉층 2의 위와 같은 연결을 반복하여 시간적으로 정렬한 형태라 할 수 있다. 이와같이 TDNN은 위층으로 갈수록 시간적인 지연요소가 증가하는데, 이러한 구조로부터 아랫층에서는 음성신호의 국부적인 특성을 그리고 위층에서는 전체적인 특성을 감지할 수 있다. 아울러 필터뱅크 계수의 차이에서 특징을 찾아내어 음성의동적인 특성을 잘 감지할 수 있다.

Ⅲ.3 새로운 시간지연 신경망 구조에 따른 diphone 인식실험

새롭게 제안한 시간지연신경망의 성능을 조사하기 위하여 diphone 인식기의 첫 단계인 위치 신경망(Position Net)을 원래의 신경망과 제안한 신경망 각각을 학습시켰다. 총 31944개의 패턴 중 반은 학습을 위해서 반은 테스트를 위해서 사용하였다. 학습은 패턴을 300, 600, 1000...으로 늘려가며 목적값과 출력값의 차가 0.1 이하가 될때까지 하였다. 학습은 약 10000회 정도 하였으며 여러 논문에서 발표된 빠른 학습 알고리듬을 도입했다. 총 학습시간은 40 MIPS 컴퓨터에서 CPU Time으로 각각 135시간, 282시간이 소요되었다.

diphone 인식기의 첫단계인 위치신경망을 대상으

표 4. 신경망 구조에 따른 인식률 비교. (a) 원래의 TDNN구조 (b) 변형된 TDNN 구조.
Table 4. The recognition rate comparison of two neural network structures. (a) Original TDNN structure, (b) Modified TDNN structure.

(a)

	Class	rate % (error/pattern)
Training	G1	91.3%(464/5324)
	G2	89.8%(539/5324)
	G3	85.9%(753/5324)
	Total	89.0%(1021/15972)
Testing	G1	84.1%(846/5324)
	G2	82.6%(927/5324)
	G3	63.0%(1969/5324)
	Total	76.6%(3742/15972)

(b)

	Class	rate % (error/pattern)
Training	G1	94.4%(298/5324)
	G2	93.5%(347/5324)
	G3	92.9%(376/5324)
	Total	93.6%(1021/15972)
Testing	G1	(590/5324)
	G2	88.9%(588/5324)
	G3	84.7%(817/5324)
	Total	87.9%(1926/15972)

표 5. 신경망 구조에 따른 특성 비교.
Table 5. The characteristics comparison of neural network structure.

Structure / Characters	New TDNN	Old TDNN	Compare
Rec. rate(training)	93.6%	89.0%	+4.6%
Rec.rate(Testing)	87.9%	76.6%	+11.3%
Computation	O(32×5×32×12)	O(16×5×616×12)	4 times
Memory	3267 weights	1651 weights	1.9 time
Training Time	282 hours	135 hours	2.1 times

로에 신경망 구조에 따른 학습 결과는 표 1에 나타내었다.

학스한 패턴에 대해서는 기존의 시간지연신경망이 89.0%의 인식률을 보이는데 반해 제안된 신경망은 93.5%의 인식률을 보였다. 그리고, 테스트 패턴에 대해서는 76.6%와 87.9%의 인식률을 보여준다.

학습한 패턴과 테스트 패턴 둘 다 제안된 시간지연 신경망의 인식률이 높게 나타났으며, 학습한 패턴에서 보다도 테스트 패턴에 대해서 제안된 시간지연신경망이 성능이 상대적으로 높은 것은 제안된 시간지연신경망이 음성의 동적인 특징을 잘 감지하고 그 특징을 연결강도(weight)에 적절히 저장했기 때문이다. 다음 표 1은 두 신경망의 특성을 정리한 것이다.

학습과 태스트 패턴에 대해서 각각 4.6%, 11.3%의 인식률이 향상되었으며, 한 패턴 당 곱하기 수(Computation)는 약 4배 정도 더 소요되며, 가중치 수는 각각 3267, 1651로 2배 정도 더 많다. 학습을 위한 최종소비 시간은 각각 282시간, 135시간으로 역시 2배 정도 더 소요된다.

Ⅳ. 실험 및 결과

본 장에서는 앞서 설명한 음성인식시스템에 대한 실험 및 결과에 대하여 이야기 한다. 그리고, 신경망 구조에 따른 음성인식 시스템의 성능을 검토하고 인식단위로서의 diphone의 장점과 diphone 분리 과정 실험에 대하여 설명한다. 그리고 오인식 수정과 어절 형성 실험에 관하여 설명한다.

Ⅳ.1 Diphone 인식기

음성 녹음은 총 1331개의 음절에 대하여 20대 남성 1명이 보통의 실험실 환경에서 Sun Sparc station에서 제공하는 Audio device를 이용하여 10번 발음한 것을 저장하였다. Sun Sparc station에서의 음성의 입출력은 8KHz sampling, 8 bit 양자화와 함께 μ-law companding을 하여 이루어지는데($\mu=225$), /dev/audio에서 나오는 데이타를 직접 저장하였다. 입력되는 신호를 $x(n)$, μ-law companding된 신호를 $y(n)$이라 할때, μ-law companding은 다음과 같은 수식에 의해 이루어진다.

$$y(n)=X_{max}\frac{\log(1+\mu\frac{|x(n)|}{X_{max}})}{\log(1+\mu)}\,sgn(x(n)), \quad (8)$$

여기서 X_{max}은 입력 신호값 중 가장 큰값을 의미하고, $sgn(x(n))$은 입력 신호의 부호를 의미한다.

음절 단위로 입력을 받은 이유는 사용 영역이 한정되어 있다면 인식률을 높이기 위해서라도 연속음성 문장을 입력받아 학습데이타로 사용해야 되지만 본 시스템에서 목표를 둔 무제한 어휘 인식을 위해서는 연속음성 문장을 입력으로 할 수 없기 때문이다. 연속음성 분장이 아닌 음절 단위로 입력을 받아 생기는 학습 데이타상의 문제점을 해결하기 위해서 인식 단위의 오인식 처리 과정을 첨가하여 연속음성 인식을 가능하게 하였다.

녹음된 음절로부터 앞, 뒤, 중간 부분을 끄집어내어 한 음절당 3개의 diphone을 만들어 저장하여 학습 및 테스트에 사용하였다.

제안된 diphone 인식 시스템은 첫번째 단계에서는 diphone을 음절 내에서의 위치 별로 3개의 그룹으로 나눈 다음 위치신경망(Position Network)를 훈련시키고, 다음 단계에서는 모음을 기준으로 첫번째 그룹을 17개의 서브그룹으로 나누어 그룹1 신경망(Group1 Network)을 훈련시키고 세번째 그룹을 9개의 서브

그룹으로 나누어 그룹3 신경망(Group3 Network)을 훈련시킨다. 마지막 단계에서는 각 26개의 각 서브그룹의 diphone을 인식하기 위하여 26개의 천이 신경망(Transition Network)을 훈련시킨다. 각각의 TDNN은 고속화 알고리듬[4]을 이용하여 훈련하였지만 방대한 가중치를 갖고 있어 30개의 신경망을 훈련시키는 40 MIPS(Million Instruction Per Second)컴퓨터를 가지고 CPU Time으로 2000시간 정도 소요되었다.

각 신경망의 인식률은 이 논문의 맨뒤에 실려있다. 위치 신경망(Position Net.)의 인식률과 위치 신경망의 각 클래스 당 인식률은 표 1에 나타나 있다.

표 6. 위치 신경망의 전체 인식률과 각 클래스 당 인식률
Table 6. Overall recognition rate and each classes recognition rate of position Net.

Data / Net.	Training rate % (error/pattern)	Testing rate % (error/pattern)
Position Net	94.7%(1693/31944)	90.1%(2371/23958)
Group1	95.5%(482/10648)	91.9%(639/7866)
Group2	94.5%(584/10648)	89.8%(806/7866)
Group3	94.1%(627/10648)	88.2%(926/7866)

학습 패턴과 테스트 패턴의 인식률은 각각 94.7%, 90.1%이다. 각 그룹은 음절의 첫 부분, 중간 부분, 끝 부분을 의미한다. 그룹3의 인식률이 다른 것에 비해 떨어진 이유는 음절 내에서 종성에 해당하는 부분이 /F, 팔, 왝,,/일 때는 중간 부분과 혼동하기가 쉽기 때문이다.

두번째 단계에 해당하는 각 그룹의 전체 인식률은 표 1에 나타나 있다.

표 7. 각 그룹의 전체 인식률.
Table 7. Each group recognition rate.

Data / Net.	Training rate % (error/pattern)	Testing rate % (error/pattern)
Total rate	88.9%(6510/5888)	84.6%(6823/44166)
Group1 Net.	89.3%(2168/20264)	85.2%(2248/15198)
Group2 Net.	92.8%(1459/20264)	87.2%(1945/15198)
Group3 Net.	84.3%(2883/18360)	80.9%(2630/1377)

학습 패턴, 테스트 패턴의 인식률은 각각 88.9%, 84.6%이다. 그룹3의 전체 인식률이 다른 그룹의 전체인식률에 비해 떨어지는 것은 그룹3 신경망에서는 17개의 모음을 9개의 모음으로 그룹핑하는 과정에서 그룹간의 특징들이 원래의 모음보다 명확하지 않기 때문이다. 각 그룹 별 클래스†당 인식률은 표 1, 표 1, 표 1에 나타나 있는데, 특히 /으/ 발음에 해당되는 diphone의 인식률이 떨어짐을 알 수 있다.

Data / Class	Traing rate % (error/pattern)	Testing rate % (error/pattern)
AA Class	91.3%(104/1192)	86.4%(123/894)
AI Class	90.8%(110/1192)	84.2%(141/894)
EO Class	85.6%(172/1192)	85.1%(133/894)
OO Class	84.2%(188/1192)	83.1%(151/894)
OI Class	93.8%(74/1192)	87.5%(112/894)
UU Class	86.2%(165/1192)	82.4%(157/894)
UI Class	85.3%(165/1192)	78.6%(191/894)
YY Class	79.4%(245/1192)	77.2%(204/894)
II Class	91.4%(102/1192)	83.9%(144/894)
YA Class	88.5%(137/1192)	83.5%(148/894)
YE Class	92.2%(93/1192)	90.5%(85/894)
YU Class	85.4%(174/1192)	81.9%(162/894)
OA Class	93.1%(82/1192)	86.1%(124/894)
YO Class	90.2%(174/1192)	87.1%(115/894)
UE Class	91.1%(106/1192)	86.8%(118/894)
IU Class	92.3%(92/1192)	89.1%(97/894)
YI Class	97.3%(32/1192)	95.2%(43/894)
Total	89.3%(2168/20264)	85.2%(2248/15198)

이것은 학습 데이타 자체가 /으/발음에 대해선 정확하지 않고, 때때로 /으/ 발음은 /의/나 /우/등의 비슷한 발음으로 발성되기 때문이다.

천이 신경망(Transition Net.)의 전체 인식률은 표 1, 표 1에 나타냈다.

이 부분의 인식률은 앞의 두 단계의 결과를 이용하기 때문에 인식률이 거의 100%에 가깝다.

† 그룹1과 그룹2의 클래스는 아(AA), 애(AI), 어(EO), 오(OO), 외(OI), 우(UU), 워(UI), 으(YY), 이(II), 야(YA), 예(YE), 여(YU), 와(OA), 요(YO), 웨(UE), 유(IU), 의(YI) 등의 17개이고, 그룹 3의 클래스는 아(AA), 애(AI), 어(EO), 오(OO), 외(OI), 우(UU), 워(UI), 으(YY), 이(II) 등의 9개이다.

표 9. 그룹2 신경망의 각 클래스 당 인식률.
Table 9. The recognition rate of group 2 neural net work.

Data / Class	Traing rate % (error/pattern)	Testing rate % (error/pattern)
AA Class	94.3%(68/1192)	89.7%(92/894)
AI Class	93.2%(81/1192)	87.8%(109/894)
EO Class	90.7%(111/1192)	83.0%(152/894)
OO Class	91.5%(101/1192)	86.7%(119/894)
OI Class	94.2%(69/1192)	89.6%(93/894)
UU Class	94.8%(62/1192)	90.3%(87/894)
UI Class	90.1%(118/1192)	83.1%(151/894)
YY Class	88.5%(137/1192)	78.7%(190/894)
II Class	92.9%(84/1192)	88.6%(102/894)
YA Class	92.5%(89/1192)	88.4%(104/894)
YE Class	93.6%(76/1192)	89.1%(97/894)
YU Class	89.9%(121/1192)	83.1%(151/894)
OA	94.4%(67/1192)	89.0%(98/894)
YO Class	92.2%(93/1192)	85.9%(126/894)
UE Class	94.4%(67/1192)	87.7%(109/894)
IU Class	93.5%(78/1192)	89.1%(97/894)
YI Class	96.9%(37/1192)	92.4%(68/894)
Total	92.8%(1459/20264)	87.2%(1459/15198)

표 10. 그룹3 신경망의 각 클래스 당 인식률.
Table 10. The recognition rate of group 3 neural net work.

Data / Class	Traing rate % (error/pattern)	Testing rate % (error/pattern)
(AA) Class	82.2%(363/2040)	78.5%(329/1530)
(AI) Class	86.5%(275/2040)	84.6%(235/1530)
(EO) Class	80.0%(407/2040)	75.1%(381/1530)
(OO) Class	82.3%(361/2040)	80.3%(301/1530)
(OI) Class	87.5%(254/2040)	83.9%(246/1530)
(UU) Class	85.3%(298/2040)	82.0%(275/1530)
(UI) Class	84.5%(316/2040)	81.4%(284/1530)
(YY) Class	85.5%(296/2040)	81.2%(288/1530)
(II) Class	84.7%(313/2040)	81.0%(291/1530)
Total	84.3%(2883/18360)	80.9%(2630/13770)

표 11. 그룹1 천이 신경망의 전체 인식률.
Table 11. The recognition rate of group 1 transition neural network.

rate / Net	rate % (train/test)	rate / Net	rate % (train/test)
AA Net	99.2%, 98.3%	YA Net	99.6%, 98.5%
AI Net	100%, 98.7%	YE Net	100%, 97.8%
EO Net	99.5%, 98.3%	YU Net	100%, 98.2%
OO Net	99.4%, 97.4%	OA Net	100%, 99.3%
OI Net	100%, 98.8%	YO Net	100%, 99.6%
UU Net	100%, 100%	UE Net	100%, 98.5%
UI Net	100%, 99.2%	YU Net	100%, 99.7%
YY Net	100%, 99.8%	YI Net	100%, 98.9%
II Net	100%, 99.6%	Total	99.85%, 98.86%

표 12. 그룹3 천이 신경망의 전체 인식률.
Table 12. The recognition rate of group 3 transition neural network.

rate / Net	rate % (train/test)	rate / Net	rate % (train/test)
AA Net	100%, 98.7%	UU Net	100%, 99.2%
AI Net	100%, 99.4%	UI Net	100%, 98.5%
EO Net	99.5% 98.3%	YY Net	100%, 99.7%
OO Net	100%, 99.4%	II Net	100%, 99.3%
OI Net	100%, 98.8%	Total	99.94%, 99.90%

여기서 유의하여야 할 사항은 신경망의 학습을 위하여 본 시스템에서는 고립 음절내에서 3개의 diphone을 끄집어 내어 이용하였기 때문에 /각찔/이란 단어는 학습 패턴에는 없는 음성이라는 것이다. 이러한 결과는 시간지연신경망을 이용한 이음소 인식기가 무제한 어휘를 인식할 수 있는 근거가 된다.

이 논문 뒤에 있는 그림 1은 본 시스템이 diphone의 위치를 찾아내 그 구간 내에서 연속적인 입력을 취하여 인식한 결과를 덤프한 것이다.

각각의 위치별 diphone 출력은 연속적인 음성 입력 중 diphone 분리 과정에 의해 선택된 입력 부분에 해당되는 출력이다. 각각 위치별 diphone 출력의 갯수는 인식 시간을 줄이기 위해 3개로 하였다.

Diphone 인식기의 출력
가 가 가
아 아 아
악 안 아

히 히 히
이 이 이
윈 윈 윈

그림. 9. /각철/이란 음성의 구간 별 diphone 인식 결과.
Fig. 9. Diphone recognition result string.

V. 음성 오인식 수정 및 어절 형성

이번 장에서는 diphone 인식기의 결과인 diphone 열들을 입력으로 확률적으로 가장 높은 어절을 찾아내는 방법에 대해서 소개한다. 아울러 한글의 발음상의 특징을 이용하여 diphone 오인식을 수정하는 방법에 대해서도 설명한다.

V.1 음성후처리 전체구조

Diphone 인식기의 후처리 과정은 크게 2가지 나눈다. 하나는 diphone 열로부터 어절을 만드는 어절 형성 시스템이고 다른 하나는 오인식된 diphone 열을 수정하는 오인식 수정 시스템이다. 하지만 이 두 시스템은 서로 유기적인 관계를 가지고 있다. 즉, 어절 형성 과정은 오인식 수정의 방법이자 결과이고 마찬가지로 오인식 수정은 어절 형성의 전처리가 된다. 음성 후처리 과정은 그림 1에 잘 나타나 있다.

음성인식 추레 단계는 크게 diphone 후보 만들기(Make Diphone Candidates), 음절 후보 만들기(Make Syllable Candidate), 어절 형성하기(Select Optimal Eojeol)등의 3단계로 이루어진다. 첫번째 단계에서는 diphone 인식기의 출력을 3개의 diphone 그룹별로 출력을 정리하고 각 그룹에서 diphone 후보를 추출해 낸다. 두번째 단계에서는 3개의 diphone 그룹 내에서 diphone을 하나씩 뽑아내어 음절을 만들어 본다. 이때, 한글 음절 중 발음이 가능한 것은 그 수가 제한되어 있으므로 음절발음사전을 이용하여 diphone을 조합하여 만들어진 음절이 발음가능한

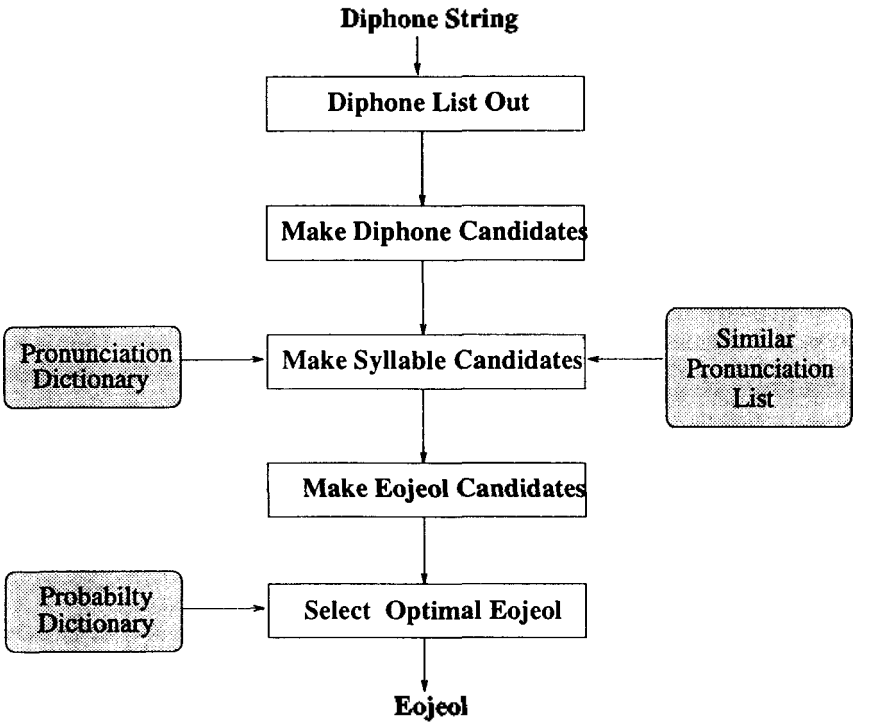


그림. 10. Diphone 인식기 후처리 시스템 구조.
Fig. 10. The post-processing structure of diphone recognizer.

지를 알아낸다. 만약에 만들어진 음절이 발음이 불가능하다면 주어진 diphone과 발음이 유사한 diphone 후보를 새롭게 제시해 음절 후보를 만든다. 이것은 각 diphone 마다 유사발음 후보 리스트를 만들어 이용하는 방법을 사용할 수도 있고 확률적인 처리방식을 도입하여 Dynamic Programming을 이용하여 diphone 후보들을 제시할 수도 있다. 본 시스템에서는 diphone 갯수가 한정되어 있으므로 유사발음 후보 리스트를 만들어 이용하는 방법을 사용하였다. 세 번째 단계는 음절들을 결합하여 어절을 만드는 과정이다. 이 단계는 어절 후보 제시(Make Eojeol Candidates)와 최적 어절 선택하기(Select Optimal Eojeol)의 두 단계로 이루어진다. 음절 후보들을 차례차례로 결합하여 어절을 만들고 이 어절 후보 중에서 확률적으로 최적인 어절을 찾아낸다. 이 단계에서는 연속음성을 인식하기 위해서는 각 어절의 경계가 명확하지 못하다는 사실을 염두에 두어야 한다. 즉, 문어체에서의 어절과 구어체에서의 언절온-발음의 편의상 나타나는 어절의 결합 혹은 어절의 분리 형태, 서로 다를 수 있으므로 이를 해결해주는 과정이 필요하다. 즉, 앞 단계의 결과인 음절열을 단순히 결합하였을 때 이것이 원하는 형태의 어절이 되는지 조사하고 만족 못할 시에 앞부분의 음절열과 결합해보던가 음절열을 분리하여 어절이 되는지 조사하는 연속 어절 후보 제시 과정이 있어야 한다. 이렇게 어절을 만들어가면서 음성 오인식 수정을 해나간다. 구체적인 어절 형성 과정은 다음 장에서 설명한다.

V.2 음성 오인식 수정 방법

음성오인식 수정은 음운법칙과 발음 가능성 정보에 의해 만들엔 확률 사전과 발음 사전을 이용하여 diphone열을 가지고, Viterbi Alogorithm을 이용하여 최적의 어절을 찾으면서 이 결과에 따라 오인식 수정을 수행하는 것을 골간으로 한다.

음성오인식 수정을 위해서 음향정보 등을 반영한 음소 천이 확률 사전을 구성하여 한 어절내의 음소열이 확률적으로 그럴듯한 음소열인지를 판단하여 음성오인식을 검색하여 수정 후보를 제시하는 시스템이 필요하다[1] 이러한 오인식 수정 후보 중에서 언어처리를 통해서 적절한 어절 등을 찾아내는 것이 음성오인식 수정의 골간이다.

본 시스템의 후처리 과정 중 오인식 수정을 하는 부분은 diphone으로부터 으절을 만드는 과정과 음절로부터 어절을 만드는 과정 두 곳에서 일어난다.

첫단계 오인식 수정은 음절의 발음가능성과 유사발음 목록을 이용하여 입력 diphone 열을 조합하여 올바른 음절을 만드는 것이다. 본 시스템에서는 음절의 발음가능성 여부를 조사하기 위해서 상용 음절발음사전(Pronunciation Dictionary)을 구성하여 diphone의 조합과 비교해보는 방법을 택했다. 유사발음목록(Similar Pronunciation List)은 사용자의 지식과 휴리스틱을 이용하여 사용자가 직접 만든다. 사람이 들어서 잘 구분이 안가는 음성을 찾기는 쉬운 일이므로 약간의 노력만 기울인다면 이 목록을 만들기는 어렵지 않다.

두번째 단계 오인식 수정은 음절로부터 어절을 형성할 때 음절열을 가지고 어절을 만들 수 있는지 검사해 본다.

구체적인 과정은 다음과 같다. 처음으로 해야 할 일은 음운 동화 현상은 음소 단위에서 일어나므로 음절열을 음소단위로 나누는 것이다. 이때 모든 음절은 3 음소로 나눈다. 이렇게 하면 어절 내에서 처리해야 될 에러가 대치에러 하나로 준다. 예를 들면 /갑엽따/는 /가엽따/의 첨가에러가 있는 경우인데 모든 음절은 3 음소의 결합이라 생각하면 이 에러는 대치에러로 바뀐다. 첨가에러나 대치에러는 상태수를 바꾸는 결과를 초래하므로 Dynamic Programming 기법으로 구현하기 어렵고 다루어야 할 에러가 많게 되므로 오인식 수정률이 떨어지게 된다. 하지만 모든 음절을 3개의 음소로 간주하면 그러한 어려움은 없어지므로 많은 이득을 얻을 수 있다. 여기에서 사용되는 음소는 초성 19개, 중성 음소 17개, 종성 음소는 /,괎,, 웩,,,/과 받침이 없는 경우의 /F/ 등 8개 이렇게 총 44개의 음소이다. 중성 음소가 17개인 이유는 /애, 에/, /왜, 외, 웨/ 중성을 각각 하나의 음소로 간주하기 때문이다.

이렇게 만들어진 음소열을 입력으로해서 각 음소들의 천이 확률을 이용하여 여러 어절 후보 중에서 적정 어절 후보를 찾아내고 음성인식기의 채널특성을 나타내는 관찰확률을 함께 이용하여 최종 어절을 찾아나낸다.

여기서 사용되는 확률적인 방법은 Viterbi 알고리듬[14]으로서 아래와 같다.

$$\text{Observation} : \vec{x}^{\,\text{def}} = x_0, x_1, x_2, \cdots, x_n, x_{n+1},$$

$$\text{State} : \vec{z}^{\,\text{def}} = z_0, z_1, z_2, \cdots, z_n, z_{n+1},$$

Letters(first 19, second 17, third 8) : $L = L_1, L_2, \cdots, L_{43}, L_{44}$.

$$\begin{cases} \textit{Given} & \vec{x},\ p(\vec{x}|\vec{z}),\ p(\vec{z}) \\ \textit{Find} & \hat{\vec{x}} \\ \textit{Such that} & \hat{\vec{x}} = argmax_{\vec{x}}\, G_1(\vec{x}, \vec{z}) \\ G_1(\vec{x}, \vec{z}) = \sum_{i=1}^{n+1} \log p(x_i|z_i) + \log p(z_i|z_{i-1}) & \\ \textit{Consider} & \text{d = depth of serach, state transition constraint.} \end{cases}$$

Viterbi 알고리듬은 언어처리기에서 통계적인 방법을 구현할 때 많이 쓰이고 있다. 언어처리기에서는 통계적인 방법을 이용하여 문법을 대치하는 것과 마찬가지로 음성오인식 수정기는 통계적인 방법을 이용하여 음운규칙을 대신하고 다음 발음을 예측하고 검증한다. 언어처리기의 입력은 단어나 구이지만 오인식 수정기의 입력은 음소나 음절이 된다.

오인식 수정 방법으로서 확률적인 접근 방식을 도입하기 위해서는 한글의 발음특징 및 제한 요건들을 음선 데이타베이스에서 통계적인 방법을 이용하여 확률값을 끄집어 내는 것이 선결 요건이 된다. 여기서 구하는 확률값은 각 음소간의 천이 확률과 각 음소당 유사 발음 확률이다. 이러한 통계적인 확률값이 의미가 있기 위해선 음성 데이타베이스가 우선 충분해야 되며 어느 정도 사용 영역에 대한 정보를 알고 있어 그 영역 고유의 발음 성격을 반영할 수 있으며 더욱 좋다. 지금까지 나온 모든 음성인식 시스템은 한정된 사용영역을 가지고 있다.

오인식 수정에 쓰이는 천이확률과 관찰확률은 구어체 문장을 입력해서 그 안에 있는 음소들의 천이 경우 수를 헤아려 얻어낸다. 한편, 음소들의 천이는 어절이 몇 음절로 구성되는가에 의해 많이 좌우되므로 음절수 별로 천이확률과 관찰확률을 구할 필요가 있다. 하지만, 현재의 알고리듬으로는 음절수를 정확하게 미리 알 수 없으므로 음절수 별로 확률을 구하는 것은 비효율적이다. 만약, 음절수를 정확하게 알 수 있다면 이를 이용하면 오인식 수정도 정확해질 것이다.

본 시스템에서는 천이 확률을 얻기 위해서 국민학교 국어책과 한글 발음 대사전[19]을 참조하였다. 관찰 확률은 한국어 음절 입력 시 diphone 인식기의 출력과 각 음소의 발음상의 특징을 이용하여 얻어냈다.

V.3 어절형성시스템

음성인식기의 출력인 diphone 열로부터 구어체의 어절형태를 끄집어내는 것이 어절 형성의 목표이다. 여기서 고려해야 될 사항은 diphone 열 자체가 에러가 포함되어 있으므로 단순히 diphone를 결합하여 어절을 만들기는 불가능하다.

이 문제를 해결하기 위해서 제안할 수 있는 방법은 출력된 diphone 열과 미리 준비한 각 어절당 diphone 리스트들 사이의 해밍 디스턴스를 구해 그 값이 가장 작은 어절을 찾아내는 방법을 들 수 있다. 그러나 이러한 방법은 어절과 언절의 차이를 해결할 수 없고, 무제한 어휘 인식 시스템에서는 해밍 디스턴스가 같은 어절들이 무수히 많을 수 있으므로 정확성이 떨어지며 비효율적이 된다. 그래서 제기된 것이 diphone으로부터 음절로 바꾸는 과정을 거쳐 우선 오인식 수정을 거친 후, 이 음절로부터 어절을 만드는 방법이다. 앞에서도 언급한 바와 같이 어절 형성은 오인식 수정과 유기적인 관계에 있다.

V.4 음성인식 후처리 결과

그림 1과 같은 diphone 출력을 입력으로하여 오인식을 수정하고, 문장의 의미를 알기 위한 첫단계로서 어절을 형성하는 것이 본 실험의 목표이다. 음성 인식 후처리의 전반적인 과정이 그림 1에 나타나 있다.

이 논문 뒤에 있는 그림 1은 /각절/을 입력했을 때 후처리되는 과정을 보여 준다.

자세한 과정은 1에 잘 나타나 있다. 음절 발음 사전과 유사 발음 사전을 이용하여 diphone 후보들에서 음절 후보를 선출한다. 이 음절 후보들을 결합하여 어절 후보를 만들고, 각 후보 어절에 대해 미리구한 관찰 확률과 천이 확률을 참조하고 Viterbi 알고리듬을 이용하여 오인식 수정을 한다. 그림 1에 각 후보 어절 별 오인식 수정 결과가 나와 있다. 한편, 오인식 수정을 거친 어절과 각 후보 어절의 천이 확률을 이용하여 최적의 어절을 찾아 내다.

그림 1에서 보듯이 오인식 된 diphone 열도 한글의 음운규칙을 반영한 확률적인 방법으로 수정되는 것을 알 수 있다. 임의로 작성된 600여개의 어절들에 대해 오인식 수정을 통해 diphone 인식률이 11.3% 향상되었다. 만약에 음소나 diphone 레벨이 아닌 어절이나 단어 레벨의 언어규칙을 사용하게 된다면 더욱

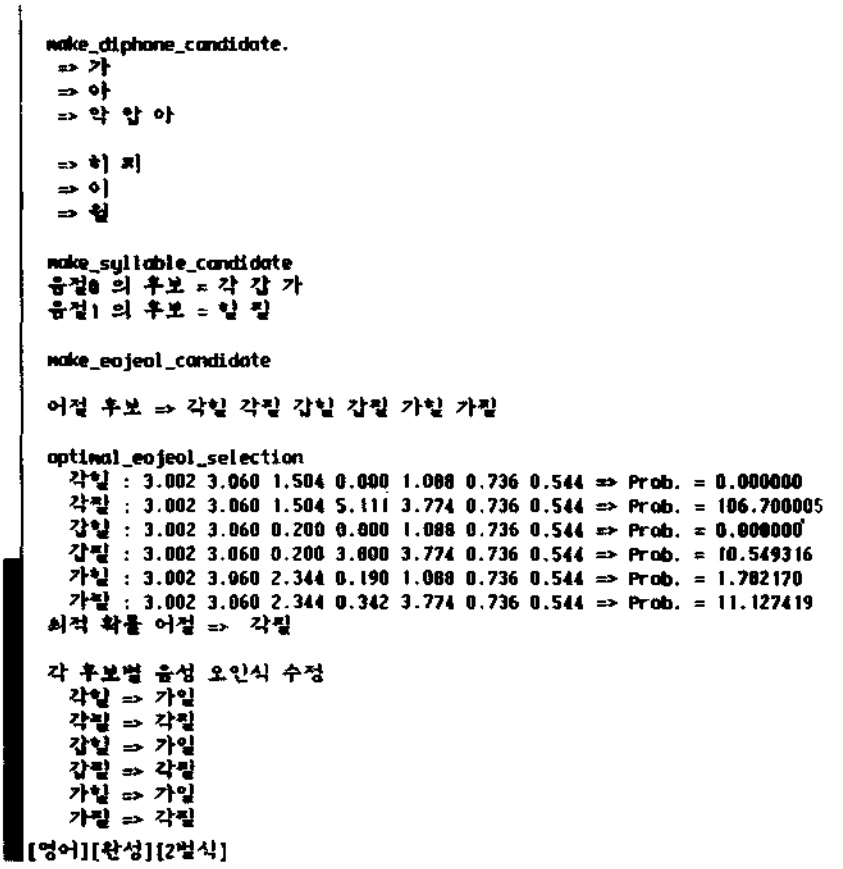


그림. 11. /각질/이란 음성인식의 후처리 과정.
Fig. 11. Post processing result.

나은 음성인식을 기대할 수 있을 것이다. 이러한 결과를 종합하여 볼 때, 음성 인식 후처리를 통해 좀 더 나은 음성인식기를 구현하여 이것을 상용화 할 수 있는 가능성을 엿볼 수 있으며, 언어처리 기술의 도입이 어느 정도 가능하여 음성 이해 시스템 구현이 그리 어렵지 않게 될 것이다.

Ⅵ. 결 론

본 논문에서는 지금까지 신경망을 이용한 diphone 인식기와 음성인식 후처리에 대해 설명해왔다. 사용한 신경망은 시간지연신경망(Time-Delay Neural Network)으로서 time shift invariance 특성과 음성의 시간적인 동적변화(time warping)을 감지하는데 탁월해 현재 이를 이용한 음성인식 시스템이 활발히 연구되고 있는 실정이다.

본 논문에서 구현한 시스템은 329개의 diphone을 인식해 후처리 과정을 거쳐 무제한 어휘를 인식할 수 있다. 그러나 분류하려는 클래스의 수가 329개로 너무 많아 단일 신경망으로 diphone을 인식하기는 힘들므로 시간지연신경망을 3층의 계층적 구조로 만들어 각 신경망의 크기를 줄이는 방법을 사용하였다. 총 신경망의 갯수는 각 층별로 1개, 3개, 26개이며, 각 층의 신경망은 같은 입력을 사용하여 각 층마다 출력을 내보낸다. 각 층의 입력은 크기가 20msec인 음성구간 16개에 대해 각각 16차의 필터 뱅크 계수를 계산해 시간지연신경망의 입력으로 사용하고, 각 층의 출력은 각각 그룹 인덱스, 모음 인덱스, diphone 인덱스에 해당한다. 이러한 3개의 출력을 취합하여 다코딩 과정을 거쳐 diphone을 출력한다. 각 층의 전

해적인 인식률은 각각 91%, 85%, 99%이며 시스템의 diphone 인식률은 77%이다. 그러나, diphone 열 출력에서 가장 빈도수가 높은 diphone 하나만 선택할 때 각 층의 전체 인식률은 92%, 89%, 99%이며, 시스템의 diphone 인식률은 81%이다.

한편, 329개의 diphone의 독특한 특징을 감지하기 위해서 기존의 시간지연신경망의 구조를 바꾸었다. 변형된 구조의 신경망은 기존과 마찬가지로 주파수별 에너지의 분포에서 특징을 감지할 뿐만아니라 주파수별 에너지의 차이에서도 특징을 감지한다. 이런 구조적인 변형에 의해 7% 정도의 인식률 향상을 가져왔다. 하지만 인식 시간과 학습시간을 많이 소비하는 단점이 있다.

음성인식의 후처리 과정은 오인식 수정과 어절 형성이라는 두가지 과정으로 분류한다. 오인식을 수정하기 위해서 음절발음사전, 유사발음 목록과 확률적인 음운규칙 등이 필요하다. 그 구체적인 과정은 다음과 같다. 우선 diphone을 결합하여 음절을 만들고 이 음절의 발음 가능성을 살펴보아 발음 불가능하다면 유사발음 목록에서 적절한 음절을 찾아낸다. 그런 다음, 이 음절들을 결합하여 후보 어절을 구성하고 확률적인 음운규칙 즉, 어절을 구성하고 있는 음소들 간의 천이 확률과 diphone 인식기의 시스템 특성을 모방하는 관찰확률을 이용하여 오인식 수정을 수행한다. 이렇게 하여 각 후보 어절마다 오인식 수정기를 거친 새로운 어절을 출력한다. 어절 형성 시스템은 한글의 음운규칙을 확률적인 방법으로 모방하여 후보 어절 중에서 가장 최적인 후보 어절을 찾아내어 출력한다.

본 음성인식 후처리기는 diphone 열을 입력받아 확률적으로 최적인 어절을 찾는 시스템으로서 앞으로의 더욱 많은 연구가 기대된다. 본 시스템은 오인식 수정을 통하여 약 11% 정도의 음성인식의 향상되어지고, 어절 형성 과정을 통하여 구문론이나 의미론 등의 언어처리 기술의 도입을 가능하게 되었다.

한편, 본 시스템의 음성 데이타베이스는 8 KHz 샘플링과 8비트의 resolution을 갖는 것으로서 기타 여러 음성인식 시스템에서 사용한 음성 데이타베이스에 비해 음질이 현격히 떨어진다. 그러나, 자동통역 전화나 음성 다이얼링 시스템 등 음성인식 시스템이 상품화가 되기 위해선 일반 전화회선의 음질과 비슷한 수준의 데이타베이스에서 음성인식이 가능하여야 하므로 인식률의 저하에도 불구하고 위에 언급한 음성 데이타베이스를 사용하였다.

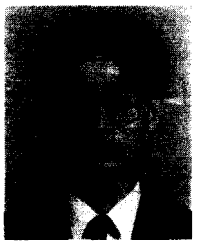
앞으로의 연구로서는 구어체의 어절을 문어체의 어절로 변환시켜주는 과정을 첨가해 기존의 언어처리 알고리듬과 방법을 그대로 적용할 수 있게 하는 연구와 실시간 인식을 위해 신경망을 하드웨어로 구현하려는 연구와 음성의 시간적인 동적 특성을 잘 반영하는 특징을 추출하기 위한 신호처리 연구가 기대된다. 아울러, 시간 지연신경망의 구조적인 변경과 학습 알고리듬을 개선하여 인식률을 더욱 높이는 연구가 필요하다. 그리고, 음성인식의 후처리과정으로서 언어처리 기술을 도입한다.

참 고 문 헌

1. L. R. Bahl, P. V. Souza, P. S. Gopakakrishnan, D. S. Kanevsky, and D. Nahamoo, "Constructing Candidate Word Lists Using Acoustically Similar Word Groups," *IEEE Trans. on Signal Processing*, vol. 40, No. 11, November 1992, pp. 2814-2816.
2. S. B. Davis and Paul Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentence," *IEEE Trans. on Acoustic and Signal Processing*, vol. ASSP-28(4), August 1980, pp. 3570366.
3. B. A. Dautrich, L. R. Rabiner, and Thomas B. Martin, "On the effects of varying filter bank parameters on isolated word recognition," *IEEE Trans. on Acoustic Speech and Signal Processing*, vol. ASSP-31(4), August 1983, pp. 793-807.
4. S. E. Fahlman, "Faster-learning variations on back-propagation : An empirical study," *Proc. 1988 Connectionist Models Summer School*, 1989, pp. 38-51.
5. J. A. Freeman and David M. Skapura, *Neural Networks : Algorithms, Applications, and Programming Techniques*, Addison Wesley, 1991.
6. J. B. Hampshire II and A. Waibel, "A novel objective function for improved phoneme recognition using time-delay neural networks," *Proc. of IJCNN*, vol. 1, 1989, pp. 235-241
7. J. B. Hampshire II and Barak Pearlmutter, "Equivalence Proofs for Multi-Layer Perceptron Classifiers and the Bayesian Discriminant Function," *Proc. 1990 Connectionist Models Summer School*, 1991, pp. 159-172.
8. J. B. Hampshire II and A. Waibel, "The Meta-Pi Network : Building Distributed Knowledge Representations for Robust Multisource Pattern Recognition," *IEEE Trans. on Patt. Anal. Machine Intell.*,

vol. 14, No. 7, July 1992, pp. 751-769.
9. B. H. Juang, L. R. Rabiner, and J. G. Wilpon, "On the use of bandpasss filtering in speech recognition," *IEEE Trans. on Acoustic Speech and Signal Processing*, vol. ASSP-35(7), July 1987, pp. 947-954.
10. L. Lamel, "A Improved Endpoint Detector for Isolated Word Recognition," *IEEE Trans. on Acoustic Speech and Signal Processing*, vol. ASSP-28(4), August 1981.
11. K. J. Lang and A. Waibel, "A Time-Delay Neural Network Architecture for Isolated Word Recognition," *Neural Networks*, vol. 3, pp. 23-43, 1990.
12. R. P. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, April 1987, pp. 4-22.
13. H. Sawai, A. Waibel, P. Haffner, M. Miyatake and K. Shikano, "Parallelism, hierarchy, scaling in time-delay neural networks for spotting Japanese phoneme/CV-syllables," *Proc. of IJCNN*, vol. 2, 1989, pp. 29-32.
14. Rajjan Shinghal, and G. T. Toussaint, "Experiments in Text Recognition with the Modified Viterbi Algorithm," *IEEE Trans. on PAMI*, vol. PAMI-1, April 1979, pp. 131-140.
15. J. I. Takami and S. Sagayama, "A Pairwise Discriminant Approach to Robust Phoneme Recognition by Time-Delay Neural Networks," *Proc. of ICASSP*, vol. 1, 1991, pp. 81-84.
16. A. Waibel, T. Hanazawa, G. Hiton, K. Shikano, and K. J. Lang, "Phoneme Recognition Using Time-Delay Neural Network," *IEEE Trans. on Acoustic Speech and Signal Processing*, vol. ASSP-37 (3), March 1989, pp. 328-339.
17. A. Waibel, "Modular Construction of Time-Delay Neural Network for Speech Recognition," *Nerual Computation*, vol. 1, No. 1, Spring 1989.
18. A. Waibel, T. Hanazawa, K. Shikano, "Modularity and Scaling Large Phonemic Neural Networks," *IEEE Trans. on Acoustic Speech and Signal Processing*, vol. ASSP-37(12), December 1989, pp. 1888-1898.
19. 한국방송공사, "표준 한국어 발음 대사전," 어문각, 1993.

▲김 경 선 1967년 8월 4일생



1991년 8월 : 포항공대 전자공학과 졸업
1994년 2월 : 포항공대 대학원 전자공학과 졸업
1994년 2월 ~ 현재 : 삼성종합기술원 기반기술연구소 연구원

▲정 홍(Hong Jeong)
현재 : 포항공과대학교 전자전기공학과 부교수