한국어 음성 데이타베이스의 저장 구조와 검색 기법

송 군 섭, 박 영 배
명지대학교 전자계산학과

The storage structure and retrieval mechanism for korean speech database

요 약

기존의 데이타베이스에 음성 데이타를 저장하여 음성 데이타 베이스를 구축하고자 할 경우, 음성 데이타의 특성이 가변장(variable length)이며, 튜플(음소 단위)의 길이가 매우 긴 패턴 데이타이므로 기존의 데이타베이스 시스템에서는 지원할 수 없다. 또, 현재의 음성 인식 시스템에서는 패턴 데이타를 순차적인 검색 방법으로 검색하고 있어 빠른 검색 방법이 요구된다.

본 논문에서는 음성 데이타를 음소 단위로 인식하기 위해 음소 패턴 데이타를 저장하고, 유사한 특성을 갖는 부류와 음소 길이에 의한 분류를 혼합한 방법을 이용하여 빠른 시간에 검색을 할 수 있게 하기 위한 저장 구조와 검색 알고리즘을 제시한다.

I. 서 론.

인간과 컴퓨터 사이에서 가장 효과적인 통신수단으로 여겨지는 음성 데이타를 체계적으로 저장하여 음성 분석과 음성 인식을 위해 효율적으로 검색하기 위한 음성 데이타베이스의 구축과 그 관리 시스템이 절실히 요구되고 있다.

현재까지 제시된 음성 데이타베이스는 주로 음성인식 전용 데이타베이스로서, 화자, 단어, 음절, 음소, 원시 음성 신호(파형) 등 모든 음성 데이타를 각각 별개의 화일로 유지하는 시스템 [1]과 화자, 단어, 음절, 음소의 데이타만 음성 데이타베이스의 테이블에 저장하고 원시 음성 신호(파형) 데이타는 별도의 화일로 유지하여 이들 사이를 포인터로 연결하여 사용하는 시스템이 있다[2]. 결국 데이타베이스 시스템과 화일 시스템을 혼용하는 상태이다.

원시 음성 신호를 디지탈 데이타로 변환하여 일련의 처리과정 후 이를 직접 데이타베이스에서 이용할 수 있는 DBMS가 구축되어야 할 것 이다.

음성 데이타는 아날로그 데이타를 디지탈 데이타로 변환하고 변환된 디지탈 데이타의 양이 너무 많아서 이를 압축하여 패턴 데이타로 데이타베이스에 저장한다. 음성 데이타는 화자에 따라 발성 지속 시간이 다른 관계로 이러한 패턴 데이타는 음소마다 각

각 프레임의 수가 가변이고 동일 음소라도 발음 환경에 따라 역시 프레임 수가 달라진다. 기존의 DBMS구조에서는 이를 지원하지 못하는 실정이다. 또한, 기존의 DBMS에서는 수치나 문자를 1:1로 매칭(matching)되는 것만을 검색하지만, 음성 데이타는 각 음소마다 프레임이 가변장인(Variable Length) 패턴 데이타 중에서 가장 유사한 패턴의 음소를 검색해야 하는 것이 바람직하다.

본 논문에서는, 음성 데이타베이스에 분석된 음성의 특징 파라미터들을 프레임 단위로 파형의 특징만을 추출한 선형 예측 계수(LPC)를 기억장소에 저장하기 위한 저장구조와 질의 되어지는 미지(未知)의 패턴에 가장 유사한 패턴 데이타를 추출하는 검색 알고리즘을 제시한다.

II. 음성 데이타베이스 요건.

2.1 음성 데이타의 특성

음성 데이타는 종래의 DBMS가 처리하는 데이타의 형식(Type)이나 접근 방법과는 매우 다르다. 이러한 음성 데이타의 특성은 다음과 같다.

첫째, 음성 데이타는 이질 형(Heterogeneous type) 데이타를 포함한다. 음성 데이타인 아날로그(Analog) 신호에서 디지탈(Digital) 신호로 변환된 원시 음성 신호(파형) 데이타를 그대로 저장하기에는 기억 장소가 많이 소요되므로 데이타량을 줄이기 위해서 프레임 단위로 파형의 특징만을 추출한 선형 예측 계수 (LPC)를 1~15차로 계산한 값(패턴)을 저장하는 방법을 많이 사용하고 있다. 또 모든 음소는 음소의 종류에 따라 또 발음 환경에 따라 프레임의 수가 달라진다. 따라서 한 음소는 여러 개의 프레임으로 구성되므로 하나의 튜플(하나의 음소)에는 프레임 단위로 여러 개의 프레임을 저장하는 중첩 릴레이션 형태가 되어야 한다.

둘째, 음성 데이타를 검색하기 위해서는 새로운 검색 알고리즘이 필요하다. 기존의 DBMS는 완전히 일치되는 수치나 문자만을 검색하지만, 음성 데이타에서는 가장 유사성이 높은 패턴 데이타로 검색해야 하므로 검색 알고리즘이 달라야 한다.

2.2 음성 데이타베이스의 논리 구조

음성 데이타를 추출하기 위하여 한 화자는 수백개의 단어와 음절들을 발음하고, 한 단어는 여러 음절로, 한 음절은 여러 음소로, 한 음소는 여러 개의 프레임으로 분석되어 데이타베이스의 각 테이블에 저장한다.

<그림 1>은 단어 '가'와'게'에 대한 프레임 테이블의 한 예로서 하나의 프레임(fr-lpc)에는 15차수까지 계산된 수치 값이 하나의 애트리뷰트에 저장된다. 또 음소 엔티티와 프레임 엔티티와의 관계(rp)는 1:n 으로서 하나의 음소는 여러 개의 프레임으로 구성되므로 각 음소당 프레임 수 만큼 <그림 1>과 같이 비-1정규형 릴레이션인 중첩 릴레이션(Nested Relation)이 된다. 이 예에서 보는 바와같이 처음의 자음 'ㄱ'은 2개의 프레임, 모음 'ㅏ'는 4개의 프레임, 다음 자음'ㄱ'은 3개의 프레임으로 동일 음소('ㄱ')라도 프레임 수가 다르게 저장되어 있음을 보여 준다[3].

ph-key	fr-no	frame unit LPC (fr-lpc)			
		lpc01	lpc02	· · ·	lpc15
ㄱ	1	-0.45803	0.141503	· · ·	0.069980
	2	-0.62025	-0.18078	· · ·	0.271163
ㅏ	1	-0.37768	0.536387	· · ·	0.040611
	2	-0.37968	0.532465	· · ·	0.047048
	3	-0.25722	0.310579	· · ·	0.245247
	4	-0.66319	-0.24501	· · ·	0.125670
ㄱ	1	-0.22011	-0.35430	· · ·	0.167092
	2	-0.58886	-0.38185	· · ·	0.239618
	3	-0.68569	-0.11010	· · ·	0.238974
ㅔ				· · ·	

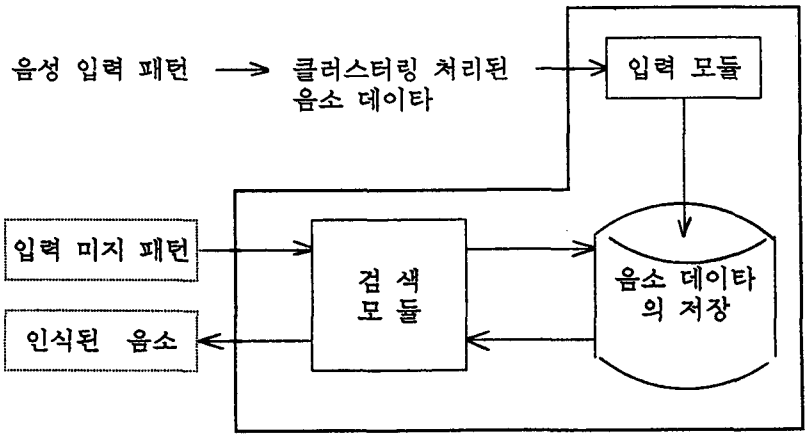
<그림 1> 프레임 테이블의 예

III. 음성 데이타베이스의 저장 구조 시스템.

3.1 음소 저장 및 검색 시스템의 구성.

본 논문에서 처리하는 음소 패턴 데이타의 저장 및 검색구조를 <그림 2>에 나타냈다. 음성의 특징 파라메터를 추출한 음성 입력 패턴을 클러스터링(clustering) 처리를 하여 유사한 부류의 특성을 갖는 음소를 같은 그룹[6]으로 구성되어진 패턴을 초기 입력으로 하였다.

클러스터링 되어진 음소 패턴 데이타는 입력모듈을 통해 저장 구조에 저장하고, 이 저장된 음소 패턴 데이타의 인덱스 패턴과 질의 되어지는 미지의 입력 패턴과의 DP-matching을 적용해 가장 유사하게 검색되어지는 음소를 출력하는 것이 전체 구성이다.



<그림 2> 음성 데이타베이스의 저장 구조 시스템.

<그림 2>에서 굵은 선 안의 부분이 본 논문에서 다루는 내용이다.

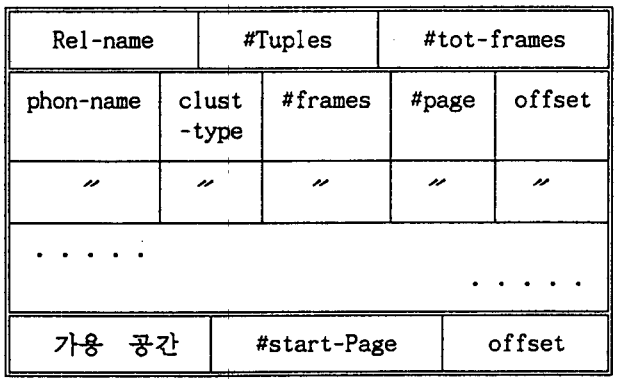
3.2 음소 패턴 데이타 저장 구조.

하드웨어 기술의 급진적인 발전으로 인해 기억장치의 용량과 처리 속도는 향상되고 가격은 저렴해지고 있다. 그러므로 데이타베이스에서의 데이타 저장은 큰 어려움을 갖고있지 않다.

일반적으로, 음성 데이타는 아날로그 신호인 파형을 디지탈 신호로 변환하여 프레임 단위로 파형의 특징만을 추출한 선형 예측 계수를 1~15차로 계산한 패턴을 저장하는데, 모든 음소는 음소의 종류와 발음 환경에 따라 프레임 수가 달라진다. 일반적으로 음소가 모음일 경우 프레임의 수가 약 70개 정도이고, 자음일 경우 24개 정도이기 때문에 음소가 모음이냐 자음이냐에 의해서도 프레임 수의 차이가 많으므로, 고정적(fixed)인 자료처리로 음소 데이타의 저장시 데이타 양의 증가에 따라 상당한 양의 저장 공간을 낭비하게 된다. 이를 해결하기 위해서는 각각의 음소가 차지하는 프레임 수만큼의 저장 영역을 갖게끔 가변적(variable)인 화일 구조가 필요하다. 한편, 각 음소의 각각의 프레임은 1~15차라는 고정된 영역의 저장 공간을 차지한다. 그러므로 한 음소는 고정된 영역의 선형 예측 계수들의 모임인 프레임으로 이루어진 가변길이의 프레임들을 동시 지원하는 저장 구조가 되어야 한다.

본 논문에서는 음소 패턴 데이타의 저장 구조 형태를 <그림 3>, <그림 5>와 같이 헤더 부분(header part)과 데이타 부분(data part)으로 구성하였다[5].

헤더 부분



<그림 3> 음소 데이타의 헤더 부분 표현

헤더 부분은 음소 패턴 데이타가 저장되어 있는 데이타 부분에 관한 정보를 저장하고 있고, 음소의 검색시 필요한 인덱스의 구성시 쓰이게 된다.

헤더 부분에 있어서, Rel-name은 릴레이션의 이름을 나타내는데, 실제적으로 '자음'이나 '모음'이 된다. #Tuples는 이 헤더에 속한 튜플(음소)의 수를 가지고,

#tot-frames는 이 헤더에 속한 총 프레임의 수를 가진다. 각각의 음소 데이타에 대한 정보를 가지는 튜플에 대한 헤더 부분의 정의는 다음과 같다.

phon-name	:	음소 명.
clust-type	:	클러스터링되어진 음소가 속한 부류.
#frames	:	한음소가 가지는 프레임 수.
#page	:	저장공간의 시작페이지 번호.
offset	:	저장공간의 시작페이지에서 실제 시작위치까지의 오프셋.

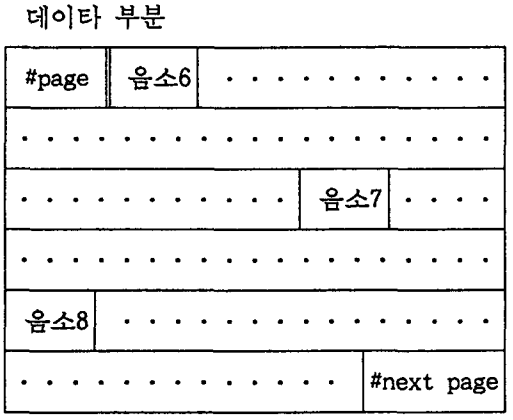
음소 데이타의 저장시 튜플단위로 저장을 하고, 그 정보는 헤더에 의해 관리가 되는데, 헤더 부분의 마지막 부분에는 차후 새로운 튜플을 저장하기 위해 저장되어 있는 음소 데이타가 차지하고 있는 기억공간을 제외한 저장 가능한 기억장소를 가진다. 즉,

가용공간	:	저장되어 튜플이 차지하는 공간을 제외한 부분의 크기.
#start-page	:	가용공간의 시작 페이지 번호.
offset	:	실제 시작위치까지의 오프셋.

을 가진다.

헤더 부분의 한 예가 <그림 4>에 보였다.

데이타 부분은 <그림 5>와 같이 페이지 단위로 저장이 되는데, 음소 데이타가 저장이 되어 할당된 저장 페이지의 영역을 채우게 되면 #next page에 다음 페이지에 대한 포인터를 두어 연결 저장하게 한다.



<그림 5> 데이타 부분 표현

자 음	n		#tot-frames	
음소 1	1	12	s	offset
음소 2	1	13	s	offset
음소 3	1	15	s+1	offset
음소 4	1	17	s+1	offset
음소 5	1	17	s+2	offset
음소 6	2	21	s+2	offset
음소 7	2	17	s+3	offset
음소 8	2	19	s+3	offset
음소 9	2	13	s+4	offset
음소 10	2	11	s+4	offset
음소 11	3	16	s+4	offset
음소 12	3	22	s+5	offset
음소 13	3	15	s+5	offset
음소 14	3	14	s+6	offset
음소 15	3	13	s+6	offset
음소 16	3	18	s+7	offset
음소 17	4	27	s+7	offset
...	...	...	...	...
음소 n	m	29	xx	offset
가용 공간		#page		offset

<그림 4 > 헤더 부분의 표현 예

VI. 검색 기법.

4.1 음소 인식.

일반적으로 음소 패턴 데이타를 인식할 때 음소 패턴 데이타의 불확정적인 요소들로

인해 다른 음소 패턴 데이타와의 매칭을 명백하게 결정지을수 없다. 즉, 저장 되어 있는 음소 패턴 데이타와 질의 되는 미지의 패턴과는 기존의 데이타베이스에서처럼 1:1 매칭에 의한 검색은 어렵다. 음성은 저장 되어 있는 음소 패턴 데이타와 질의되는 패턴이 같은 음소라 하더라도 발성시의 조건들(발성자, 잡음의 정도, 발성 속도 등)에 따라 패턴의 형태가 달라지기 때문이다.

본 논문에서는 이러한 패턴 매칭을 위해 DP-matching[4]을 음소 패턴 데이타의 검색에 적용하였다.

4.2 검색 알고리즘

음소인식의 정확도를 높이기 위해서는 가능한 많은 데이타(저장되어 있는 음소 패턴 데이타)와의 dp-matching이 요구된다. 가장 높은 정확성을 기하기 위해서는 저장되어 있는 모든 개개의 음소 튜플과 순차적인 dp-matching을 하여 그 결과 가장 가까운 거리 값을 가지는 음소를 선택하는 것이다. 하지만, 정확도를 높이기위해 dp-matching을 하는 데이타량이 많을 수록 처리 속도는 떨어진다.

음소 패턴 데이타는 그 특성상 다음과 같이 상반된 요소를 가질 수 있다.

1. 같은 길이의 튜플이라도 내용이 다를 수 있다.
2. 다른 길이의 튜플이라도 내용이 같을 수 있다.

일반적으로, 음소 패턴 데이타는 불확실한 요소를 가지는 데이타 형태를 가지면서 데이타량은 방대하므로 그 검색기법이 어려운 실정이다.

본 논문에서는 저장되어 있는 음소 패턴 데이타와 질의 되는 미지 패턴과의 빠른 검색을 하기위한 방법으로 특성상 유사한 부류의 음소군[6]으로 분류된 음소 패턴 데이타에서 비슷한 길이의 범위(bound)에 속한 음소 부류로 다시 분류하는 혼합 방법을 이용한 검색 구조를 제시한다.

음소 데이타의 특성상 유사한 부류의 구분은 <그림 3> 헤더 부분의 clust-type에 의해 분류되고, 비슷한 길이의 범위에 속한 음소 부류의 분류는 일정 범위 즉, 자음의 경우에 있어서는 각 음소의 프레임이 최소 2개에서 최대 48개 사이의 프레임 수를 가지므로 범위를 프레임의 수가 1~7개 시 1, 8~14개 시 2, 15~21개 시 3, 22~28개 시 4, 29~35개 시 5 등으로 구분되어 주어진다. 이와같이 하여 <그림 4> 헤더의 내용으로부터 추출하게 된다. 이때 인덱스의 구성은 <그림 6>과 같이 된다.

phon-name	clust-type	frm-bnd-type	#page	offset

<그림 6> 인덱스의 구성 형태

여기서, frm-bnd-type은 프레임의 길이에 의해 분류된 값을 가지게 된다. <그림 4>의 예에 적용을 해보면 같은 clust-type이면서 프레임의 길이에 의한 분류에서도 같은 부류에 속하는 음소가 존재한다. 이러한 음소의 인덱스 추출은 무작위 방법에 의해 그 중 하나를 추출하기로 한다. <그림 4>에 대한 인덱스 추출의 예가 <그림 7>에 나타나 있다.

음소 2	1	2	s	offset
음소 4	1	3	s+1	offset
음소 6	2	4	s+2	offset
음소 8	2	3	s+3	offset
음소 9	2	2	s+4	offset
음소12	3	4	s+5	offset
음소13	3	3	s+5	offset
음소14	3	2	s+6	offset

<그림 7> 추출된 인덱스의 예

이와같은 인덱스 추출의 방법은 모음에 대해서도 범위를 달리하여 인덱스를 구성할 수 있다.

사용자가 검색하고자 하는 미지의 패턴을 입력했을 때 먼저 그 입력된 음소 프레임의 길이를 계산하여 1 ~ 48개 사이에 있다면 자음 부분의 인덱스를, 40 개 이상의 프레임 수를 가진다면 모음 부분의 인덱스를 선택하여 dp-matching을 시도한다. 만일, 프레임의 수가 40개 ~ 50개 사이에 있어 자음과 모음의 선택이 애매한 경우에는 자음과 모음의 인덱스를 차례로 매칭하는 것으로 한다.

이상의 알고리즘을 정리하면 다음과 같다.
미지의 패턴이 입력되어 매칭을 요구하면,

(1) 입력된 음소 패턴 데이타의 프레임 수를 계산하여 검색할 자음, 모음의 인덱스 또는 모두 선택할 것인가를 결정한다.

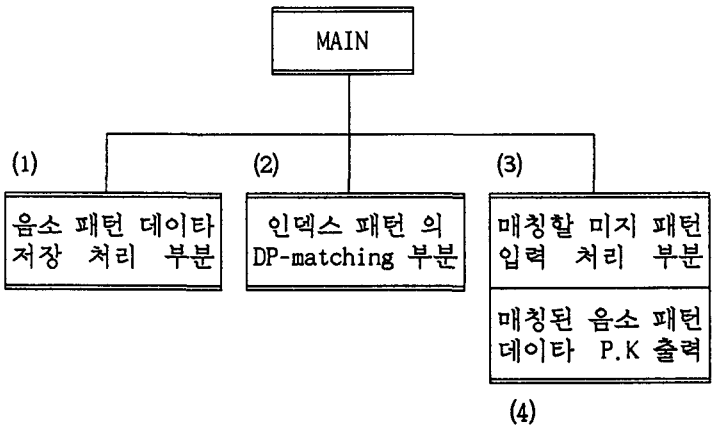
(2) 입력된 미지의 패턴과 선택된 음의 인덱스들과의 dp-matching을 적용하여 거리값을 계산한다.

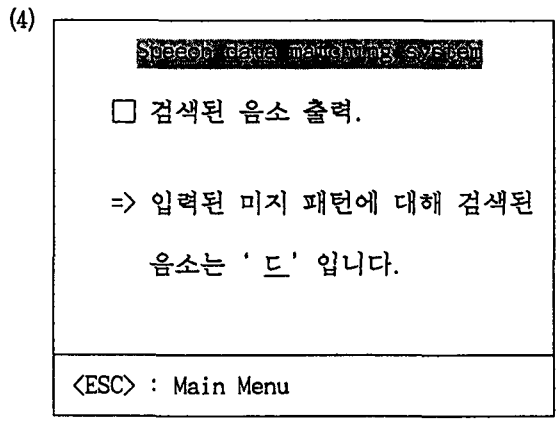
(3) 가장 가까운 거리값을 가진 인덱스를 선택하고, 헤더 부분에서 동일한 clust-type에 같은 frm-bnd-type에 속한 음소들을 탐색한다.

(4) 탐색된 동일 clust-type을 갖는 음소들에 대하여 다시 dp-matching을 적용한다.

(6) (5)의 결과로 가장 가까운 거리 값을 가지는 음소(phon-name)를 출력한다.

실제로 구현하기위한 모듈은 <그림 8>과 같이 4 개의 모듈로 구성이 된다. 여기서, 모듈 (4)는 검색하고자 하는 미지 패턴의 결과 값 처리부이다.





<그림 8> 전체 모듈 구성 및 화면 설계

V. 결 론.

본 논문에서는 음성 데이타베이스 시스템에서 음성 데이타를 저장하고 검색하는 데 있어서 방대한 저장 용량을 차지하는 가변 길이 애트리뷰트를 가진 음성 데이타를 페이지 단위로 할당된 저장공간에 저장하기 위한 방법과 기존의 DBMS의 검색 방법과는 데이타 특성이 다른 음성 데이타를 검색하기 위해 동일한 특성을 갖는 부류와 음소 길이에 의한 분류를 혼합하여 기존의 순차적인 검색에 의한 방법보다 빠른 검색을 할 수 있는 검색 방법을 제시하였다.

앞으로의 연구 과제는 본 논문에서 제시한 저장·검색 기법과 기존의 방법과의 성능

평가를 하는 것이고, 나아가 실제적으로 음성 입력에서 인식까지 하나의 시스템을 구현하는 것이 되겠다.

참 고 문 헌

[1] K. Akiba, T. Irumano, et al, "Speech Database for Research of Japanese Speech Recognition, Spring Metting Acoustic Soc. Japan, paper 1-4-22, March, 1982.

[2] H. Kuwabara, K. Takeda, et al, "Construction of a Large-Scale Japanese Speech Database and Its Management System, " proc. ICASSP 89, 1989.

[3] 박영배, 이석호, "한국어 음성 데이타 처리를 위한 데이타베이스 시스템의 설계" Vol 17, No.2, Oct. 1990.

[4] H. Sakoe, S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", IEEE transactions on Acoustics, Speech, and Signal processing, Vol. ASSP-26, No. 1, 1978.

[5] 강현철, 박두환, 백승민, "고기능 RDBMS 프로토타입 개발", 한국정보과학회, '90 가을 학술 발표 논문집, pp. 15 ~ 18.

[6] 신미영, 김영인, 박영배, "한국어 음소 패턴 데이타의 클러스터링", 한국정보과학회, '91 가을 학술 발표 논문집, 심사중.

[7] C. Faloutsos, "Access Methods for Text", Computing Surveys, Vol. 17, No. 1 1985.

[8] Gio Wiederhold, File Organization for Data base Design, McGRAW-HILL, pp.131 -171, 1988.

[9] James Martin, Computer Data-Base Organization, Prentice-Hall, pp. 327-357, 1977.