한글 주소의 오인식 수정을 위한 효율적인 후처리 알고리즘

이 성환, 김 은순
충북대학교 전자계산학과

An Efficient Postprocessing Algorithm for Error Correction in Hangul Address Recognition

Seong-Whan Lee and Eun Sun Kim
Department of Computer Science, Chungbuk National University

요약

본 논문은 한글 주소의 오인식 수정에 관한 연구로서, 필기자의 다양한 필기 형태와 입력 장치의 특성으로 인하여 문자 인식 단계에서 발생할 수 있는 혼동을 줄이고 오류를 효율적으로 수정하는 후처리 알고리즘을 소개한다. 특히, 주소의 행정 구역부에 대해서는 정합해야 할 문자 모델의 범위를 줄여줌으로써 높은 인식률과 처리 속도를 기록하였으며, 문자 인식의 결과에 임계값과 백트랙킹 방법을 도입한 후처리 알고리즘을 적용하여 더욱더 높은 인식률을 나타낼 수 있었다. 번지부와 건물부에 대한 오인식도 제안된 각각의 알고리즘을 적용함으로써 효과적으로 수정할 수 있었다.

우리나라의 25,000여 행정 구역을 바탕으로 작성 가능한 주소들 중에서 임의의 150개 주소 데이타에 대하여 제안된 후처리 방법을 포함한 다양한 후처리 방법으로 실험한 결과, 행정 구역부에 대하여 98%이상의 높은 인식률을 보임으로써, 제안된 후처리 알고리즘이 효과적임을 알 수 있었다.

I. 서론

폭발적으로 증가하는 정보의 양을 효율적으로 관리하고자 하는 노력이 대두됨에 따라 필기체 문자 인식 기술에 대한 연구는 더욱 가열되고 있다. 그러나 최근까지 진행되어 온 필기체 문자 인식 기술에 관한 연구 결과는 필기자의 다양한 필기 형태와 입력 장치의 특성으로 인하여 높은 인식률과 처리 속도를 기대하기는 어렵다. 따라서 이러한 한계성을 극복하기 위해서는 문자 인식 단계에서 발생하는 오인식을 실시간에 효율적으로 수정할 수 있는 후처리 알고리즘의 개발에 관한 연구가 절실하다[민병우91].

본 논문은 일상 생활에서 자주 사용되는 주소를 후처리 대상으로 하는데, 효과적인 주소의

후처리를 위하여 입력 문자와 정합해야 할 문자 모델의 수를 줄이는 방법을 제시하였으며, 주소의 각 부분별 특성에 따라 후처리 알고리즘을 개발함으로써 문자 인식 단계에서 발생 할 수 있는 오인식을 효과적으로 수정하였다.

본 논문의 구성은 다음과 같다. II 장에서는 종래에 연구된 주소의 오인식 수정을 위한 후처리 알고리즘을 소개하며, III 장에서는 행정 구역명들과 더불어 번지, 통, 반 등의 구체적인 정보와 아파트, 학교 등의 건물명이 나타나는 우리나라 주소의 일반적인 형태를 고려하여 각기 특성에 따라 적합하게 개발된 후처리 알고리즘을 소개하고, IV 장에서는 제안된 후처리 알고리즘을 포함한 다양한 후처리 알고리즘들의 성능을 비교한다. 마지막으로 V 장에서는 본 연구에 대한 결론과 함께 앞으로의 연구 방향을 제시한다.

II. 관련 연구

종래에 연구된 주소의 오인식 수정을 위한 후처리 방법을 살펴보면, 국내의 경우는 관련 연구를 전혀 찾아볼 수 없으며, 국외의 경우는 일본 주소의 오인식 수정을 위한 후처리 방법에 대하여 3 편의 연구 결과가 발표되었는데, Izaki 등[Izaki83]은 입력된 주소의 각 문자에 대하여 특징을 추출하고 이들의 특징을 각 표준 문자의 특징과 비교하여 거리가 가장 작은 것을 후보 문자로 출력하면, 이 후보 문자들의 순위에 따라 가중치를 두어 사전의 행정 구역명과 문자 정합을 함으로써 가장 유사한 후보 주소를 찾는 방법을 소개하였고, Suzuki 등[Suzuk90]은 후보 문자 출력 과정에서 후보 1 순위의 문자가 얻은 값과 더불어 후보 1 순위와 후보 2 순위의 문자가 얻은 값의 거리를 함께 고려하여 인식의 정확도를 검사한 후 오류의 여지가 있는 문자는 소수의 후보 문자들을 출력하여 이 후보 문자들과 사전의 행정 구역명을 정합하는 방법을 제시하였다. 또한 행정 구역 사전을 이용할 수 없는 번지의 처리에는 오류 규칙 테이블(error rule table)과 연결 행렬(connection matrix)을 이용한 후처리 방법을 적용하였다. 그리고 최근에 Marukawa 등[Maruk91]은 연속적으로 필기된 주소의 처리를 위하여 오토마타를 도입하여 행정 구역을 분할한 후 행정 구역명 사전과 문자 정합을 하는 방법을 소개하였다.

이러한 주소의 오인식 수정을 위한 후처리 방법들은 입력된 주소의 각 문자들을 모든 표준 문자와 정합함으로써 출력되는 소수의 후보 문자들을 이용하여 후처리하는 방법이다. 이와 같은 방법은 인식 대상이 주소라는 점을 감안할 때, 매우 광범위하고 불필요한 정합을 수반하므로 비효율적이다. 필기체 한글 주소 인식에 있어서 종래의 방법을 적용할 경우, 현행 KS C 5601 완성형 코드 체계는 무려 2,350 종류의 문자를 가지고 있기 때문에 비교의 횟수는 2,350번에 이른다. 한자의 경우는 훨씬 많은 문자 집합을 가지고 있어 문자 인식의 혼동 범위가 더욱 크다. 이러한 비효율성을 개선하기 위하여 본 논문에서는 행정 구역 사전을 이용하여 주소에 나타날 수 있는 문자 만을 특히, 주소를 이루고 있는 문자들의 각 위치에 나타날 수 있는 문자 만을 정합 대상으로 삼는다. 본 논문에서 제안되는 후처리 방법이 상기 방법들과 다른 또 하나의 특이한 점은 주소를 구성하고 있는 각 문자에 대한 문자 인식을 완전히 끝낸 후에 후처리를 하는 것이 아니라, 먼저 한 계층의 행정 구역에 대하여 인식과 후처리를 하여 그 계층에 대한 행정 구역명을 결정한 다음, 이 행정 구역명을 바탕으로 그 다음 계층을 처리한다는 점이다.

III. 한글 주소의 오인식 수정을 위한 후처리 알고리즘

3.1 우리나라 주소의 특성

우리나라의 행정 구역은 그림 1과 같이 계층적인 구조로 이루어져 있다.

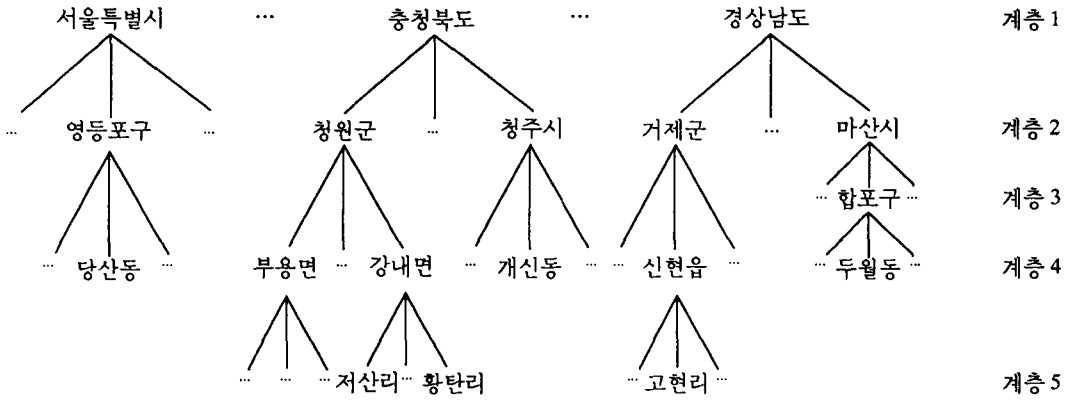

그림 1. 우리나라 행정 구역의 구조

그림 1에서 알 수 있듯이 우리나라 행정 구역 구조상 특이한 점은 계층 2의 하위 계층이 읍, 면, 또는 동이 아닌 구(이하 '예외구'라함)로 분류되는 행정 구역이 존재한다는 것이다. 경기도 부천시, 경상남도 마산시 등 5개 지역이 여기에 해당된다. 이러한 행정 구역의 구조를 바탕으로 필기된 주소의 예는 다음과 같다.

경상남도 마산시 합포구 두월동 주공아파트 가동 206호 (예외구를 포함하는 주소)

충청북도 청주시 개신동 산 48번지 충북대학교 전자계산학과 (일반적인 주소)

위와 같이 하나의 주소를 구성하는데 있어서 가장 먼저 나타나는 행정 구역명은 특별시, 직할시 또는 도를 나타내는 계층 1에 대한 것이고, 그 이하의 행정 구역명은 계층 1을 시점으로 하여 계층적으로 구성되어 있음을 알 수 있다. 또, 이렇게 구성된 행정 구역명들 뒤에는 번지, 통, 반 또는 아파트, 학교 등의 건물명이 나타난다. 본 연구에서는 이러한 주소를 효과적으로 처리하기 위하여 각 부분별로 각기 다른 후처리 알고리즘들을 적용하는데, 이들 각 부분에 대한 명칭을 다음과 같이 정의한다. 위에서 예시한 주소처럼 행정 구역명들이 나열된 부분을 행정 구역부라 정의하고, 번지 및 통, 반이 기술된 부분을 번지부라 정의하며, 건물명들이 나열된 부분을 건물부라 정의하였다.

3.2 행정 구역 사전의 구축

행정 구역 사전은 계층적인 구조로 이루어진 우리나라 행정 구역의 구조를 바탕으로 그림 2와 같이 5개의 사전으로 구성된다. 행정 구역 사전 1은 계층 1(특별시, 직할시, 도)에 관한 것이고 행정 구역 사전 2는 계층 2(구, 군, 시)에 관한 것이며 행정 구역 사전 3은 계층 3(예외구)에 관한 것이다. 행정 구역 사전 4에는 계층 4(읍, 면, 동)에 관한 정보가 수록되어 있고 행정 구역 사전 5는 계층 5(리, 동)에 관한 정보가 수록되어 있다.

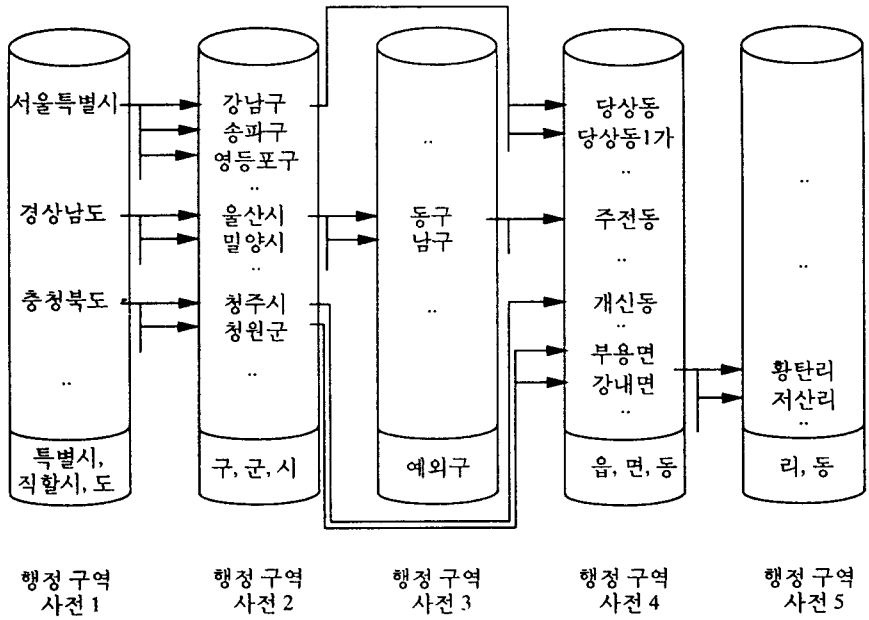

그림 2. 행정 구역 사전의 구성

우리나라에서는 우편물 분리의 편리성을 도모하기 위하여 공공 기관 및 규모가 비교적 큰 건물에 대하여 계층 4에 해당되는 하나의 우편 번호를 부여함으로써 건물을 임의의 행정 구역명으로 다루고 있기 때문에 본 논문에서 사용하는 행정 구역 사전 4에는 읍, 면, 동 뿐만 아니라 건물명도 수록하여 이러한 건물명을 처리하고 있음은 물론, 사용자가 이용 목적에 따라 우편 번호를 갖지 않는 건물명도 행정 구역 사전에 추가하여 수록할 수 있도록 행정 구역 사전이 설계되어 있다.

이렇듯 건물명을 행정 구역 사전에 입력할 수 있도록 하는 것은 사람에 따라서 주소를 기재하는 방법이 상이한 점을 고려할 때 대단히 편리한 방법을 제공하게 된다. 나아가, 주택추세가 아파트 단지화되고 있는 점을 비추어 볼 때, 이러한 아파트에 대해서도 행정 구역 사전에 수록함으로써 이용의 편리성 내지는 극대화를 꾀할 수 있다.

3.3 행정 구역부의 후처리

본 논문에서는 행정 구역부의 문자 인식을 시작하기에 앞서, 인식하여야 할 주소 단어를 이루고 있는 각 문자의 위치에 발생 가능한 문자들을 행정 구역 사전을 이용하여 추출한다. 앞 절에서 소개된 주소의 예에 대하여 설명하면 먼저 '충청북도'의 각 자리에 나타날 수 있는 후보 문자를

행정 구역 사전을 이용하여 추출하게 되는데 주소 단어 '충청북도'의 '충'의 자리에 나타날 수 있는 문자는 '강(강원도), 경(경기도, 경상북도, 경상남도), ..., 제(제주도)'와 같은 15개의 문자들이고 '청'의 위치에 나타날 수 있는 문자는 '원(강원도), 상(경상남도, 경상북도), ..., 주(광주직할시, 제주도)'이다. 이상에서 알 수 있듯이 입력 문자인 '충'을 인식하는데 있어 현행 KS C 5601 완성형 코드 체계인 2,350개의 표준 문자와 비교하는 것이 아니라 15개의 문자 모델 만을 비교함으로써 정합 대상이 약 15/2,350로 감소된다. 문자 인식 단계[이성환92b]에서는 위와 같이 행정 구역 사전에서 추출된 문자들과 입력 문자를 정합하여 가장 유사한 문자를 순위별로 출력한다. 이 때 발생할 수 있는 오인식을 수정하기 위하여 문자 인식 단계에서 출력된 후보 문자를 이용한 후처리 과정에 들어가는데, 아래의 표 1을 이용하여 그 과정을 설명하기로 한다.

주소 단어	후 보 문 자 들				
	1 순위	2 순위	.....	n 순위	그 외
C_1	$C_1(1)$	$C_1(2)$	.....	$C_1(n)$	
C_2	$C_2(1)$	$C_2(2)$	.....	$C_2(n)$	
.	.	.	.	.	
.	.	.	.	.	
.	.	.	.	.	
C_m	$C_m(1)$	$C_m(2)$	.....	$C_m(n)$	
W	W(1)	W(2)	.....	W(n)	W(n+1)

표 1. 주소 단어 정합에 사용되는 후보 문자들과 가중치

여기서, $C_1C_2...C_m$은 m개의 문자로 구성된 주소 단어이며, $C_i(1), C_i(2), ..., C_i(n)$는 입력 문자 C_i에 대한 n 순위까지의 후보 문자를 의미하며, W(1), W(2), ..., W(n)는 순위별 가중치를 나타낸다. 사전에 수록된 하나의 주소 단어를 $S_1S_2...S_m$이라 할 때, $C_1C_2...C_m$과 $S_1S_2...S_m$의 거리는 각 문자들의 거리의 합으로 정의한다. 예를 들어, 후보 문자 $C_1(4)$이 S_1과 일치할 경우(즉, $S_1 = C_1(4)$), 후보 문자 C_1과 사전에 수록되어 있는 문자 S_1의 거리는 W(4)만큼이다. 입력된 하나의 행정 구역명과 사전의 행정 구역명과의 거리(D)는 다음의 식으로 정의할 수 있다.

$$D = \sum_{i=1}^{m} d(S_i, C_i) = \sum_{i=1}^{m} W(\alpha) \ \{\alpha : S_i = C_i(\alpha)\} \qquad (1)$$

단, $\alpha = 1, 2, ..., n$

본 논문에서 사용된 가중치 $W(\alpha)$는 α가 후보 문자의 순위를 나타낼 때 $W(\alpha) = \alpha - 1$ 로 정의하였다. 만약 S_i가 $C_i(1), C_i(2), ... C_i(n)$중 어느 것과도 일치하지 않을 경우에는 가중치를 W(n+1)로 한다.

위의 단어 정합 방법을 적용함에 있어 계층 1에 대한 행정 구역명에는 입력 문자와 같은 갯수를 가진 행정 구역명 만을 정합 대상으로 하였고, 그 이하 계층의 행정 구역은 문자의 갯수가 같거나 한 문자 더 많은 행정 구역명을 정합 대상으로 선택하였다. 그 이유는 계층별 행정 구역을 나타내는 특별한 문자(도, 시, 특별시 등)를 생략하는 사람들의 습관에 대하여 입력 문자의 수 만큼만 출력함으로써 사전을 확장하지 않고서도 해결할 수 있도록 하기 위함이다. 단, 계층 1에 대해서는 생략 범위가 크기 때문에 정식의 행정 구역명(서울특별시, 충청북도 등) 만을 기준으로 문자 정합을 하는 것은 비효율적이다. 또한 계층 1에 대한 행정 구역명의 수가 매우 적기 때문에 위의 문제점은 행정 구역명에 대한 다양한 기재 형태를 유형화하여 행정 구역 사전을 구성함으로써 쉽게 해결될 수 있도록 하였다. 구체적으로 말하여 '서울특별시'에 대해서는 '서울' 또는 '서울시'를 행정 구역 사전 1에 추가하였다.

그림 3은 임의의 주소 단어 '충청북도'에 대하여 문자 인식 단계에서 후보 문자들을 출력한 예와, 위에서 소개한 문자 정합의 결과를 보여주고 있다.

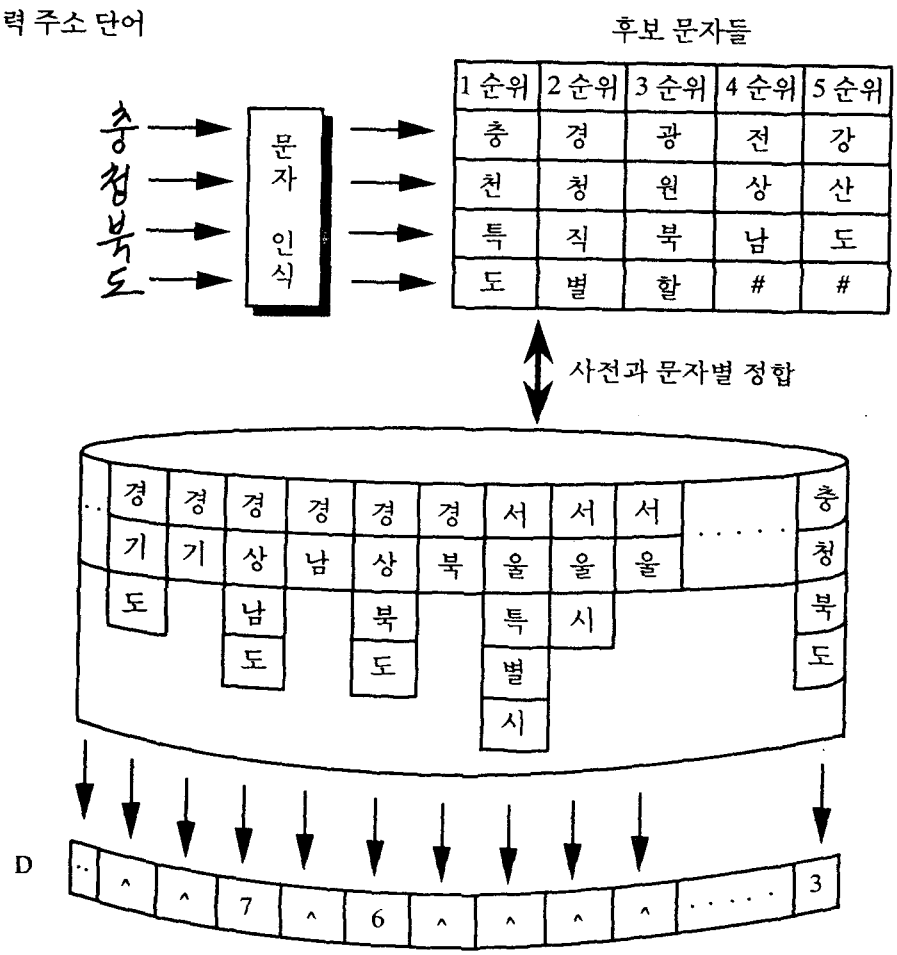

그림 3. 후보 문자들을 이용한 문자 정합

위의 그림 3에서는 먼저, 입력 문자 '충'에 대한 제 1 순위의 인식 결과는 '충'이 가장 유사한 것으로 선택되었고 그 다음 2 순위로는 '경'이며, 차후의 순위에 대해서도 이와 같은 의미로 출력된

후보 문자를 나타내고 있다. 그 다음은 행정 구역 사전과, 행정 구역 사전에 수록된 행정 구역명들과 후보 문자들과의 문자 정합에 의하여 얻어진 차이를 보여주고 있다. 이 예에서 계층 1에 대한 행정 구역명의 4 번째 자리에 나타날 수 있는 문자는 단지 '도', '특', '직'의 3개의 문자이므로 4 순위와 5 순위의 후보 문자가 출력되지 않음을 '#'로 표시하였으며 정합 대상이 아닌 행정 구역명은 '^'로 표시하였다.

이상과 같은 방법으로 후보 문자와 가장 작은 거리를 갖는 사전의 행정 구역명을 k 순위까지 결정하여 저장한다. 이때 k 순위까지의 행정 구역명을 '후보 행정 구역명'이라 하며 1 순위로 선택된 후보 행정 구역명을 바탕으로 다음에 나타나는 주소 단어에 대하여 위와 같은 방법을 반복함으로써 주소 인식은 진행된다.

그러나 1 순위로 선택된 후보 행정 구역명을 확정하여 하위 행정 구역을 처리할 경우 잘못된 결과가 초래될 수 있다. 그 이유는 정합 과정에서 사전의 각 행정 구역명이 얻은 거리가 2개 이상의 행정 구역명에 대해 동일하게 된다든지, 문자 인식의 결과로 출력된 후보 문자가 해가 아닌 다른 행정 구역명과 더 가깝게 정합되는 경우이다. 이러한 문제점을 해결하기 위하여 거리에 대한 임계값과 백트랙킹을 도입한다. 즉, 현재 처리하고 있는 계층을 계층 L이라 할 때, 후보 문자들과 사전의 행정 구역명과의 정합에 의하여 1 순위로 선택된 후보 행정 구역명이 얻은 거리가 임계값 이하이면 계층 L에 대하여 선택된 후보 행정 구역명을 바탕으로 계층 L + 1을 진행하고 그렇지 않으면 상위 계층에 대한 후보 행정 구역명들 중에서 다음 순위에 해당되는 후보 행정 구역명으로 접근하여 이를 바탕으로 계층 L을 다시 처리하는 백트랙킹을 수행한다.

이상에서 소개한 임계값의 검사와 백트랙킹을 반복적으로 수행하여도 문자 정합 과정에서 계층 L에 대하여 1 순위로 선택된 후보 행정 구역명과 문자 인식 단계에서 출력된 후보 문자들과의 거리가 임계값 이상이 계속될 경우는, 계층 L의 상위 계층인 계층 L - 1에 대한 후보 행정 구역명이 문자 정합 과정에서 얻은 거리를 함께 고려한다. 임의의 계층 L의 상위 계층인 계층 L - 1에 저장된 k 순위까지의 후보 행정 구역명을 모두 검색하기 위하여 k - 1번의 백트랙킹을 시도하였다면 두 계층에 대하여 각각 k개의 후보 행정 구역명을 얻게 되는데, 문자 정합 과정에서 각각의 후보 행정 구역명들이 얻은 거리를 합하여 그 값이 가장 작게 되는 두 계층의 행정 구역명을 계층 L과 계층 L - 1의 후보 행정 구역명으로 확정한다. 즉, d(i, j, k)가 i 번째 백트랙킹에서, 계층 j에 대한 k 순위의 후보 행정 구역명이 얻은 거리를 나타낸다고 할 때, 다음 식에서 가장 작게 되는 항목에 대한 상위와 하위 계층의 행정 구역명을 계층 L과 계층 L - 1의 후보 행정 구역명으로 정한다.

$$\text{Min}\{d(0, L-1, 1) + d(0, L, 1), d(1, L-1, 2) + d(1, L, 1), \ldots, d(k-1, L-1, k) + d(k-1, L, 1)\} \quad (2)$$

위와 같은 방법으로 계층 L의 행정 구역명이 선택되었으면 이를 바탕으로 계층 L + 1을 처리한다. 이와 같이 두 계층을 함께 고려함으로써 임의의 계층에서 사전의 각 행정 구역명에 대하여 거리가 같은 것이 2개 이상 존재하는 경우나 심지어 문자 인식 단계에서 출력된 후보 문자들이 해가 아닌 행정 구역명과 더 가깝게 정합되어 이 행정 구역명을 바탕으로 다음 계층을 처리하게 된다 할지라도 올바른 수정 결과로 유도할 수 있다.

3.4 번지부의 후처리

번지부에 대한 오인식 수정은 사전의 정보를 이용하여 수정하는 방법을 적용할 수 없다. 왜냐하면 번지부는 한정된 몇개의 문자들과 '0 ~ 9'의 숫자들의 조합으로써 구성되므로 이들의 모든 조합을 사전에 수록한다는 것은 의미가 없기 때문이다. 그러므로 번지부의 오인식 수정은 표 2의 번지부에 나타날 수 있는 문자들의 연결 가능 행렬을 이용하여 후처리를 한다.

		연결 가능한 문자								
		0	1~9	-	/	통	반	번	지	산
시작문자	0	○	○	○	○	○	○	○		
	1~9	○	○	○	○	○	○	○		
	-		○							
	/		○							
	통		○							
	반		○							
	번								○	
	지		○							
	산		○							

표 2. 번지부에 나타날 수 있는 문자들의 연결 가능 행렬

주소의 번지부에 표기 가능한 문자는 '0~9', '번','지', '통', '반', '산' , '/', '-' 등 한정된 문자들 뿐이다. 그러므로 문자 인식 단계에서는 이 문자들 만을 정합 대상으로 하여 인식을 하고 여기에서 발생하는 오인식은 연결 가능 행렬을 이용하여 수정할 수 있는데, 예를 들어 '218(반\번)지'로 인식된 경우는 표 2에 의하여 '218번지'로 수정된다.

3.5 건물부의 후처리

건물부에 나타날 수 있는 건물명들은 건물, 빌딩, 공공 기관명, 학교,아파트 등에 고유 명사가 결합된 형태이다. 이러한 정보를 빠짐없이 사전에 수록하는 것은 정보 수집의 어려움 뿐만이 아니라 방대한 기억 공간이 필요하다. 따라서 다음과 같은 건물부 후처리를 소개한다.

$S_1S_2S_3...S_n$을 n개의 문자로 이루어진 건물명이라 할 때, 건물명을 다음과 같이 이분한다. 예를 들어 입력된 건물명이 '충북대학교'라 할 때, 다음의 식에 의하여 이분된 결과는 {ϕ, 충북대학교}, {충, 북대학교}, {충북, 대학교}, {충북대, 학교}, {충북대학, 교}, {충북대학교, ϕ}이 된다.

$$S_1S_2S_3...S_n \Rightarrow \left\{ \prod_{i=1}^{k} S_i \prod_{j=k+1}^{n} S_j \right\} \qquad 0 \le k \le n, k\text{는 정수} \tag{3}$$

단, $\prod_{i=1}^{0} S_i = \phi$

위와 같이 분할된 총 (n + 1) x 2개의 항목(각각의 항목을 '부분 단어'라 정의함) 중에서 {ϕ, n}

과 {n, ϕ}은 같은 것이며 여기에서 공집합에 대한 항목은 의미가 없는 것이므로 n개의 문자로 이루어진 한 항목만을 고려한다. 그리고 (n + 1) x 2개의 항목 중에서 왼편에 하나의 문자로 이분된 항목은 고려의 대상으로 삼지 않는다. 그 이유는 왼편의 한 글자는 일반적으로 고유 명사의 형태를 취하고 있으므로 한 글자로 이루어진 고유 명사의 광범위성으로 인하여 사전 구성이 어렵고, 또한 사전을 구성한다는 것이 의미가 없기 때문이다.

건물부는 행정 구역부의 사전과 같이 건물명에 대한 정보를 수록한 사전을 이용하여 후처리를 하는데 이 사전은 건물명에 공통적으로 들어가는 일반적인 명사 즉, '빌딩', '건물' , '학교' 등과, 건물명에 자주 쓰이는 고유 명사 등을 문자의 갯수별로 분류하여 구성한다. 이러한 사전 구성의 방법은 건물명에 대한 사전 구성의 부담과 그 한계성을 극복하고 있다.

건물부에 대한 후처리 방법은 분할된 (n + 1) x 2개의 항목 중에서 위와 같은 이유로 4개의 항목이 제외된 (n + 1) x 2 - 4개의 항목에 대하여 각각의 항목들이 갖는 문자 수 만큼의 문자들로 이루어진 건물명에 대한 정보를 수록하고 있는 건물 사전과 문자 정합을 하게 된다. 문자 정합의 방법은 행정 구역부에 적용된 방법과 동일하며 그 결과에 따라 (n + 1) x 2 - 4개의 항목 중에서 가장 작은 거리를 갖는 항목을 선택한다. 이때 거리가 같은 것이 2개 이상 발생한 경우는 문자 수가 많은 항목을, 그리고 이분된 항목들 중에서 오른편에 위치한 항목을 우선 순위로 하여 선택한다.

n개의 문자로 구성된 건물 단어를 이분하여 얻은 (n + 1) x 2 - 4개의 항목에 대하여 문자 정합을 행한 결과, 오른편의 i개의 문자로 이루어진 항목에 대하여 사전에 수록된 임의의 건물 단어(이하 후보 건물 단어라 함)가 1 순위로 선택되었다고 가정하면 선택된 후보 건물 단어에 대한 출력은 다음의 방법을 따른다.

IF i개의 문자로 구성된 후보 건물 단어와 문자 인식 단계에서 1 순위로 출력된 후보 문자들 간의 거리 < 건물에 대한 임계값
THEN i개의 문자로 구성된 후보 건물 단어를 출력
ELSE 문자 인식 단계에서 1 순위로 출력된 후보 문자들을 출력

위의 항목에 대응되는 항목(위의 가정에 의하자면 왼편의 (n - i)개로 이루어진 항목을 의미함)에 대한 출력은 다음과 같다.

IF (n - i)개의 문자로 구성된 다른편의 항목에 대응되는 건물 사전에서 1 순위로 선택된 건물 단어와 문자 인식 단계에서 1 순위로 출력된 후보 문자들간의 거리 < 건물에 대한 임계값
THEN (n - i)개의 문자로 구성된 후보 건물 단어를 출력
ELSE 문자 인식 단계에서 1 순위로 출력된 후보 문자들을 출력

IV. 실험 및 결과 분석

본 절에서는 [이성환92b]의 문자 인식 결과에 각 부분별 후처리 알고리즘을 적용한 실험 결과를 소개한다. 먼저, 행정 구역부 후처리 방법의 효용성을 검증하기 위하여, 우리 나라의 25,000

여 행정 구역[채신부90]을 바탕으로 10인이 임의로 작성한 150개 주소 데이타에 대하여 본 논문에서 제안한 후처리 방법을 포함한 다양한 후처리 방법들에 대한 실험 결과를 비교하였다. 각 입력 문자에 대한 후보 문자는 5 순위까지 고려하였으며 필기된 주소의 행정 구역부를 이루고 있는 평균 문자 수는 10자이다.

본 실험에서 고려된 행정 구역부에 대한 후처리 방법들은 다음과 같다.

방법 1 : 문자 인식 단계에서 입력된 행정 구역명들을 구성하는 모든 문자에 대하여 후보 문자를 출력하면 이들 후보 문자를 이용하여 사전의 행정 구역명들과 정합하여 후처리하는 방법.

방법 2 : 한글의 특성을 이용하기 위하여 문자 인식 단계에서 출력된 후보 문자를 자소별로 분리하여 사전에 수록된 행정 구역명들과 자소별 정합하여 후처리하는 방법.

방법 3 : 본 논문에서 제안한 방법으로, 문자별 인식을 시작하기 전에 주소 단어를 이루고 있는 각 문자의 위치에 나타날 수 있는 문자를 행정 구역 사전에서 추출하고, 추출된 문자들과 입력 문자를 정합하여 얻은 순위별 후보 문자들을 이용하여 행정 구역 사전에서 후보 행정 구역명을 선택하여 후처리하는 방법.

방법 1은 [Izaki83, Maruk91, Suzuk90]에서 사용된 후처리 방법이고, 방법 2는 한글의 특성을 고려한 후처리 알고리즘으로 자소별 정합을 시도함으로써, 방법 1의 문자별 정합과 비교하여 그 효과를 실험하기 위해 제안된 방법이며, 방법 3은 본 논문에서 제안한 방법으로, 주소의 특성을 이용하여 정합해야 할 문자 모델의 수를 줄여주어 이 문자 모델 만을 대상으로 인식을 하고, 임계값과 백트랙킹을 도입한 후처리 알고리즘을 적용하여 문자 인식 단계에서 발생할 수 있는 오인식을 수정하는 방법이다.

위의 각기 다른 3 가지 방법에 대한 실험 결과를 표 3에 나타내었는데, 각각의 방법에 대하여 후처리를 적용하기 전과 후의 문자별 인식률과 단어별 인식률 및 문자별 평균 처리 속도를 비교하였다. 또한, 표 4는 번지부와 건물부에 대하여 본 논문에서 제안한 후처리 알고리즘을 적용하기 전과 후의 인식률을 비교하고 있다. 본 실험의 환경은 IBM PC 486(33MHZ) 상에서 Turbo-C 언어로 구현하였다.

	후처리 사용전			후처리 사용후		
	단어별 인식률	문자별 인식률	처리 속도	단어별 인식률	문자별 인식률	처리 속도
방법 1	45%	52%	0.17초	67%	78%	0.32초
방법 2	45%	52%	0.17초	82%	88%	0.41초
방법 3	94%	96%	0.06초	97%	98%	0.13초

표 3. 행정 구역부 주소 데이타에 대한 다양한 후처리 방법들의 성능 비교

번지부	후처리 사용전의 인식률	85%
	후처리 사용후의 인식률	92%
건물부	후처리 사용전의 인식률	74%
	후처리 사용후의 인식률	87%

표 4. 번지부와 건물부에 후처리 알고리즘을 적용하기 전과 후의 인식률 비교

위의 표 3에서 알 수 있듯이 행정 구역부에 대한 3 종류의 후처리 방법들 중에서 방법 2의 한글의 특성을 이용한 방법은 그리 효과적이지 못하며 본 논문에서 제안된 방법 3이 다른 방법들 보다 처리 속도는 물론 인식률이 월등히 뛰어남을 보이고 있다. 또한 번지부와 건물부에 대해서도 본 논문에서 소개한 후처리 알고리즘이 효과적임을 알 수 있다(표 4).

V. 결론

본 논문에서는 한글 주소 인식에 있어서, 문자 인식 단계에서 발생할 수 있는 오인식을 효율적으로 수정하는 새로운 후처리 알고리즘을 제안하였다. 특히, 행정 구역부에 대해서는 각 문자의 자리에 나타날 수 있는 문자를 행정 구역 사전에서 추출하여 정합해야 할 문자 모델의 범위를 줄여줌으로써 문자별 인식 속도를 크게 향상시킴은 물론 높은 문자 인식률을 기록하였다. 또한 여기에 임계값과 백트랙킹 방법을 도입한 후처리를 적용하여 오인식을 수정함으로써 98% 이상의 주소 인식을 실현할 수 있었다. 그 밖에 번지부와 건물부에 대해서는 각각의 특성에 따라 개발된 후처리 알고리즘을 적용하여 문자 인식 단계에서 발생하는 오인식을 효과적으로 수정하였다.

본 논문에 소개된 후처리 알고리즘은 필기체 주소 인식에만 국한되는 것이 아니고 타이핑, 인쇄 및 기타 기계 매체에 의해 기재된 주소의 인식에도 높은 인식률을 기록할 수 있으나, 문자의 형태 변형이 매우 심하여 문자 인식 단계에서 상당한 혼동과 오류가 발생하는 온라인 및 오프라인 필기체 한글 주소 인식에 그 효과를 발휘할 수 있다. 본 방법은 요즘 주목받고 있는 음성 인식의 한 분야인 음성 주소 인식에도 적용할 수 있는데 그 방법은 사전을 구성한 다음, 스피커를 통하여 입력된 음성 주소 데이타를 음절별 음성 인식 과정을 거침으로써 해결할 수 있다[이성환92a].

앞으로 연구 과제로는 주소를 필기할 때 필기자에 가해지는 제약, 예를들어 계층별 행정 구역명의 띄어쓰기 등의 제약을 완화하기 위하여 공백없이 연속적으로 필기된 주소에 대한 처리 기법과 필기자의 일반적인 습관, 예를 들어 널리 알려진 행정 구역명을 생략하는 경우를 효율적으로 처리할 수 있는 후처리 알고리즘을 개발하는 것이다.

감사의 말씀

행정 구역 사전의 설계에 많은 조언을 해 주신 민 병우씨와 주소 데이타 작성을 도와준 충북대학교 전자계산학과 김 기철, 김 영준, 김 용준, 김 진남, 박 정선, 박 희선, 서 민희, 송 희헌,이 동준, 이 승진 씨에게 감사드린다.

참고 문헌

[Izaki83] Y. Izaki, M. Nakanishi and Y. Kawasaki, "Postprocessing of Handwritten Kanji Character Recognition," Proc. of Int. Conf. on Text Processing with a Large Character Set, Tokyo, Japan, Oct. 1983, pp. 197-202.

[Maruk91] K. Marukawa, M. Koga and Y. Shima, H. Fujisawa, "An Error Correction Algorithm for Handwritten Chinese Character Address Recognition," Proc. of Int. Conf. on Document Analysis and Recognition, Saint-Malo, France, 1991, Vol. 2, pp. 916-924.

[Suzuk90] A. Suzuki, S. Miyahara and F. Obashi, "Selective Correction for Address Recognition Errors," Proc. of Int. Conf. on Computer Processing of Chinese and Oriental Languages, Changsa, China, April 1990, pp. 95-100.

[민병우91] 민병우, 이 성환, "문자 인식을 위한 오인식 수정 기술," 한국정보과학회지, 9권 1호, 1991년 2월, pp. 7-13.

[이성환92a] 이 성환, "한글 주소 인식 방법 및 장치," 특허 출원 제 15673호, 1992년 8월.

[이성환92b] 이 성환, 박 정선, "통계적 특징 추출 방법을 이용한 샘물체 필기 한글의 오프라인 인식," 제 4회 한글 및 한국어 정보처리 학술대회 발표 논문집, 서울, 1992년 10월(게재 예정).

[체신부90] 체신부, 통신 구획 편람, 1990년 5월.