

## CRM 위한 효율적인 주소 정제 시스템 개발\*

박규진<sup>○</sup> 송원문 김은주 김명원  
숭실대학교

{dugini, gtangel, blue7786}@ssu.ac.kr mkim@comp.ssu.ac.kr

### Development of Efficient Address Cleaning System for CRM

Youngjin Han<sup>○</sup> Younggun Rim  
Soongsil University

#### 요 약

최근 기업들은 인터넷, 이벤트 등의 다양한 매체를 통해 방대한 양의 고객 주소 데이터를 수집하고 있다. 이렇게 수집된 고객 주소 데이터는 DM발송 등을 통한 마케팅 및 캠페인에 활용함으로써 서비스의 질을 향상시키고, 고객의 충성도를 높인다. 그러나 사용자가 입력한 주소의 일부는 띄어쓰기가 정확하지 않은 주소, 정보가 누락된 주소, 오타가 있는 주소, 구주소 등의 입력으로 주소의 표준 형태를 갖지 않거나 일관성이 없다. 또한 2007년 4월부터 부분시행예정인 신주소 체계를 위해 기존의 주소를 적절히 변환해야만 한다. 본 논문에서는 부정확하거나 불완전한 주소를 표준 형태의 정확한 주소로 변경하는 효율적인 주소 정제 시스템을 제안하고 개발한다. 본 시스템은 먼저 입력 주소의 유니코드 분석을 통한 한글 입력 오류 교정과 해쉬테이블을 이용한 건물명의 표준 형태 변환을 수행한다. 이 과정이 완료된 주소들은 주소 트리 모델 구축을 통해 띄어쓰기가 부정확한 주소의 처리, 역주소테이블(inverted address table)을 이용한 누락 주소의 처리(enrichment), 구주소에 대한 트리 모델을 생성한 신주소 변환 처리 등의 프로세스를 통해 정제하게 된다. 실험을 위해 오타와 띄어쓰기, 누락된 정보가 있는 주소를 생성하여 타 정제 시스템과 성능을 비교하였다. 생성된 데이터를 비교한 결과 기존 주소 정제 시스템의 오류율 7.5% 보다 제안된 방법으로 개발된 본 시스템의 오류율이 0.45%로 낮음을 확인하였다.

#### 1. 서 론

최근 기업들은 DM(direct mail) 발송 등을 위해 인터넷, 이벤트 등의 다양한 매체를 통해 방대한 양의 고객 주소 데이터를 수집한다. 이렇게 수집된 고객 주소는 DM 발송 등을 통한 마케팅 및 캠페인에 활용함으로써 서비스의 질을 향상시키고 고객의 충성도를 높일 수 있는 고객 데이터로 사용된다. 사용자가 입력한 주소는 일반적으로 띄어쓰기가 정확하지 않은 주소, 정보가 누락된 주소, 주소에 오타가 있는 주소, 변경되기 전의 구주소 입력 등으로 일관성을 가지지 않는 경우가 많다. 이러한 일관성 없는 고객 주소를 사용하여 등록된 고객에게 마케팅을 수행하는 것은 과중한 비용 손실을 유발시키는 동시에 기업의 이미지를 손상시키는 요인이 된다.

본 논문에서는 사용자가 입력한 일관성 없는 주소의 품질(data quality)을 높이기 위한 주소 정제 시스템인 AddressCleans을 제안한다. AddressClean은 한글 입력 오류(typing error)에 대하여 정제가 불가능한 기존 시스템의 문제점을 해결하기 위해 유니코드 분석과 한글 유사도 비교를 사용하여 한글 입력 오류를 수정한다.

본 논문은 크게 5장으로 구성된다. 2장에서는 정보통신부에서 2000년 5월 1일자로 변경 고시한 주소 체계와 개편된 우

편번호의 구성과 체계 우편번호 부여 방법에 대해 설명한다. 그리고 주소 정제의 종류에 대해 간단히 기술하고 현재 주소 정제 서비스를 하고 있는 업체들의 정제 시스템을 분석한다. 3장에서는 본 논문에서 제안하는 AddressClean의 구성과 전처리 방법 그리고 띄어쓰기가 정확하지 않은 주소 정보가 누락된 주소, 오타가 있는 주소, 구주소 입력 등을 정제하는 각각의 과정에 대하여 기술한다. 4장에서는 실제 주소 데이터와 테스트를 위해 생성한 데이터들을 사용하여 제안하는 AddressClean의 효율성을 검증하고, 5장에서는 결론 및 향후 연구에 대하여 기술한다.

#### 2. 관련 연구

효율적인 주소정제를 위해서는 현재 우편번호 체계와 주소에 대한 우편번호 부여 방법 등에 대한 분석이 먼저 이루어져야한다. 본 장에서는 2000년 5월 1일자로 변경된 현 우편번호의 구성 및 체계 우편번호 부여 방법을 분석하고 2007년 4월 주소 개정된 주소 체계를 알아본다. 또한 현재 주소 정제 서비스를 하고 있는 업체들의 시스템 현황을 분석한다.

##### 2.1 현재 우편번호 체계

현재 사용하고 있는 우편번호는 1970년 7월 최초로 제정되어 2000년 5월 1일 <그림 1>과 같이 최종 개편되어 사용되고 있으며 우편번호 부여 방법은 정보통신부에

\* 본 연구는 서울시 산학연 협력사업 (10581cooperateOrg93111)의 연구결과로 수행되었음

서 개시한 '우편번호 찾는 법'을 기준으로 부여된다. 발송용 번호는 지역별 구분을 위한 번호이며 배달용 번호는 배달 업무를 원활하게 하기 위해 부여되는 번호로서 기본적으로는 수취인 주소에 다량배달처 건물명이 있는 경우와 다량배달처 건물명이 없는 경우로 나누어진다[1].

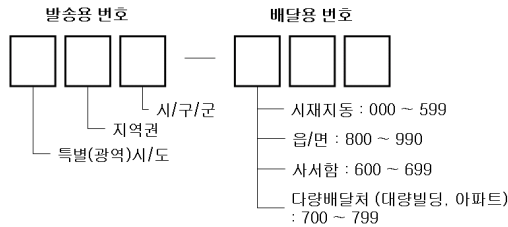


그림 1. 우편번호 구성 및 체계

● 다량배달처 건물명이 있는 경우

수취인 주소에 건물명이 있는 경우 그 건물명이 다량배달처에 해당하는지 확인하게 된다. 다량배달처 우편번호가 별도로 부여된 건물인 경우 지번단위 우편번호에 우선하여 다량배달처 우편번호를 기재한다

표 1. 다량배달처 건물명이 있는 경우의 예

예) 서울특별시 강남구 논현동 4번지 거평빌딩 302호 - 거평빌딩의 올바른 우편번호: 135-726 (○) - 행정동(논현1동)으로 잘못 기재: 135-011 (×) - 법정동(논현동)으로 잘못 기재: 135-010 (×) - 지번(4번지)으로 잘못 기재: 135-811 (×)
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<표 1>에서 거평빌딩은 다량배달처로 구분되어 있으므로 '135-726'이라는 우편번호를 부여해야 정상적인 우편배달이 가능해진다. 만약 거평빌딩을 다량배달처로 구분하지 않으면 그 상위 주소의 우편번호를 부여하게 되어 정확한 우편 번호를 부여할 수 없다

● 다량배달처 건물명이 없는 경우

건물명이 있더라도 다량배달처 건물이 아닌 경우 또는 수취인 주소에 건물명이 없고 주소지 번지만 있는 경우에는 그 번지에 해당하는 지번단위 우편번호를 기재하며, 읍/면 지역은 리단위 우편번호를 기재한다 다음과 같은 <표 2>의 경우 현우빌딩은 다량배달처로 구분되어 있지 않은 건물이므로 그 상위 주소인 지번단위로 우편 번호를 부여하여야 한다

표 2. 다량배달처 건물명이 없는 경우의 예

예) 서울특별시 강남구 논현동 4번지 현우빌딩 101호 - 논현1동 4번지의 올바른 우편번호: 135-811 (○) - 행정동(논현1동)으로 잘못 기재: 135-011 (×) - 법정동(논현동)으로 잘못 기재: 135-010 (×)
--------------------------------------------------------------------------------------------------------------------------------------------

2.2 도로명 주소 체계

2007년 4월 5일부터 '도로명 주소 표기'에 관한 법률 시행에 따라 주소 관리 방법이 변경되었다 이는 현 주소 체계가 일체의 토지 조사 사업의 결과 수탈 및 조세

징수를 목적으로 부여한 토지 지번 방식으로 급격한 경제 개발과 인구 급증으로 인한 토지 이용 다변화로 종래 방법을 그대로 사용하는데 한계가 있기 때문이다 도로명 주소 표기 방법은 다음과 같다

- 현재 주소 체계: 서울시 동작구 상도5동 송실대학교
- 도로명 주소 체계: 서울시 동작구 사당로50 송실대학교

도로명 주소 체계는 2007년 4월 현재 광역시를 중심으로 시행되었으며, 2008년까지 새 우편번호를 도입하여 2012년에 완전 변경할 예정이다 그러나 아직까지 예산 준비 부족으로 광역시를 제외한 다른 지역에서는 새 주소 시행이 확정되지 않고 있고 중복 도로명이 많은 등 여러 가지 문제가 있다. 그러므로 본 논문에서는 도로명 주소 체계를 이용한 주소정제는 고려하지 않는다

2.2 주소 정제 시스템

2.2.1 주소 정제 시스템의 기능 개요

주소 정제 시스템이란 입력된 주소에서 오류 정보를 검출하고 정형화된 주소 정보로 변화하여 올바른 우편번호를 부여할 수 있도록 하는 시스템이다 따라서 우편번호와 표준 주소 형태를 같이 저장관리하는 기능을 포함한다. 이러한 주소 정제 시스템은 일반적으로 <표 3>과 같은 기능들을 가진다.

표 3. 주소 정제 시스템의 기능 개요

기능	설명
주소 표준화	<ul style="list-style-type: none"> <li>○ 생략된 다량배달처 표현을 복원 표기</li> <li>○ 동일 공동 주택이나 빌딩에 대해 여러 형태의 명칭으로 표현된 것을 우편번호에서 제공하는 다량배달처의 명칭을 기준으로 변경</li> <li>○ 지번을 한 가지 형태로 표기</li> </ul>
주소 정형화	<ul style="list-style-type: none"> <li>○ 주소 구성 요소 세분화</li> <li>○ 주소 구성 요소 간격 정리                             <ul style="list-style-type: none"> <li>- 행정명칭 간격 조정</li> <li>- 건물명칭과 부속층의 간격 조정 및 구분 표기</li> </ul> </li> </ul>
Enrichment	<ul style="list-style-type: none"> <li>○ 번지 추가 기능</li> <li>○ 동, 리 명 추가 기능</li> <li>○ 동, 리 명 오타 수정 기능</li> </ul>
신 우편번호 변환	<ul style="list-style-type: none"> <li>○ 지번그룹 우편번호 변환 기능</li> <li>○ 리 우편번호 변환 기능</li> <li>○ 다량배달처 우편번호 변환 기능</li> </ul>

<표 3>에서 주소 표준화는 입력된 주소에 우편번호에서 제공하는 다량배달처의 명칭이 있는지를 검사하고 교정하는 기능을 말한다. 그리고 주소 정형화는 주소를 쓰는 형식에 관련된 사항으로 입력된 주소의 형태가 올바른지를 판단하고 수정하는 기능이다 또한 Enrichment는 세부적인 주소를 판단하는 기능이며 신 우편번호 변환은 번지를 더 세분화하여 우편번호를 부여하는 기능이다

2.2.2 기존 주소 정제 시스템

기존 주소 정제 시스템은 수지원넷[2], 이스턴 테크놀로지[3], 공영DBM[4], OPENmate[5] 등에서 개발되었으나, 한글 입력 오류나 한글 오타 누락 주소에 대한

\* 1일 배달물량 500통 이상인 대형 빌딩 및 기관 아파트 등 공동주택  
 † 행정동은 효율적인 행정 운영을 할 수 있게 더 세분화한 동  
 ‡ 법정동은 호적과 같은 공식 문서상에 표기되는 땅이름



하위 시/군/구가 존재한다. 이때, “경기 고양시 일산동구 마두1동”의 경우처럼 도 하위에 시/군/구에 해당하는 필드가 “고양시 일산동구”와 같이 두 개로 나누어지면 L2와 L22로 구분하여 노드를 생성한다 L3에는 L2의 각 시/군/구가 갖는 하위 읍/면/동이 존재한다. 주소 트리의 루트에서 모든 단말노드인 LE까지의 경로에는 반드시 L1에서 L3의 노드를 가지며, 그 외의 경우 리/도서, 다량배달처, 산번지 등 번지의 존재에 따라 각각의 노드가 추가 된다.

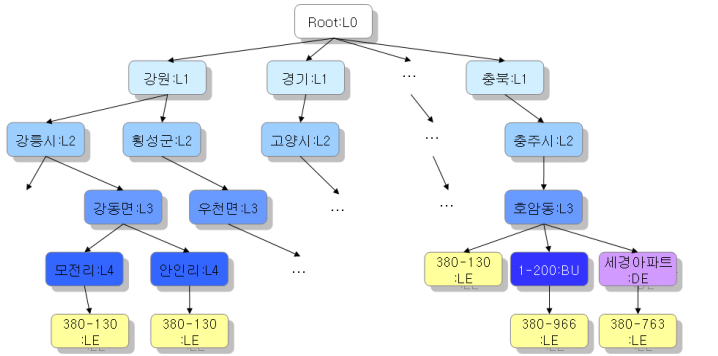


그림 4. 주소 트리

검색하여 단일 노드가 검색되었을 때 우편번호를 부여한다. 만약 우편번호를 부여하지 못한다면 주소 정제 엔진을 이용하여 주소가 정제되게 된다. 주소 정제 엔진에서는 정확한 주소를 찾고 자주 나타나는 유형의 오류 주소를 처리하여 주소 정제 과정의 효율성을 높이기 위해

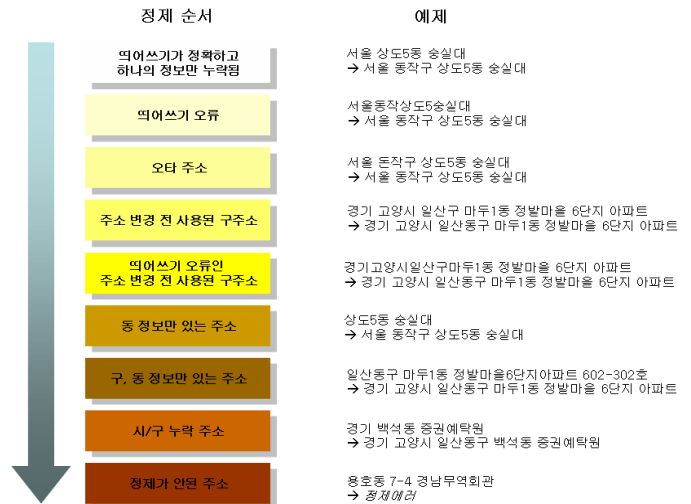


그림 5. AddressClean의 주소 정제 과정

표 6. 주소 트리의 노드 타입

노드 타입	주소 구분	노드 수	노드 타입	주소 구분	노드 수
L1	시/도	16	L5	도서	126
L2	시/군	234	DE	다량배달처	11,258
L22	구		BU	번지	
L3	읍/면/동	6,142	SA	산번지	
L4	리	17,623	LE	우편번호	

AddressClean의 파싱 모듈에서는 주소 트리를 이용하여 주소의 파싱을 수행한다. 사용자의 입력 주소의 앞 토큰부터 주소트리의 노드가 갖는 주소 토큰과 비교하여 일치하는 경우, 입력 주소의 다음 토큰과 노드의 하위노드가 갖는 주소 토큰의 비교과정을 반복하여 파싱과정을 수행한다.

### 3.3 신 우편번호 변환 모듈

파싱 모듈에서 주소 트리를 이용하여 사용자의 입력주소를 파싱하고 파싱된 결과를 정형화된 주소 틀에 저장한다. 이때, 주소 파싱 과정에서 주소트리의 단말노드까지 검색된 경우 정상주소로 분류하여 신 우편번호 변환 모듈에서 우편번호를 부여하고 단말노드까지 검색이 안된 경우 오류주소로 분류하여 주소 정제를 수행한다. 신 우편번호 변환 모듈은 파싱된 정보를 사용하여 상위주소와 세부주소를 구분한다. 상위주소와 세부주소에 포함된 읍/면/동, 다량배달처, 산번지, 번지의 정보 유무에 따라 다량배달처, 번지, 행정동, 법정동 순으로 우편번호를 부여한다.

### 3.4 주소 정제 엔진

AddressClean의 주소 정제 엔진에서는 파싱 모듈에 의해 파싱된 주소를 정제한 후 주소 트리의 단말노드까지

<그림 5>와 같이 9개의 단계를 순차적으로 수행한다. 첫 번째 단계에서는 정확한 띄어쓰기 정보를 가진 주소와 주소 정보(시도, 시군구, 읍면동, 리, 도서) 중 하나의 레벨이 누락 된 경우에 대한 주소를 처리하며 주소 정제에서 이 단계가 가장 많이 사용된다. 또한 두 번째 단계에서는 첫 번째 단계 다음으로 많이 일어나는 경우로 주소에 띄어쓰기를 하지 않은 경우, 주소 트리를 이용하여 주소가 정제된다.

$$\text{유사도} = \frac{1}{3n} \sum_{i=1}^n w_i \quad (\text{식 1})$$

$n$ : 글자수  
 $w_i$ :  $i$ 번째 음절에서 정확히 일치한 자소의 수

세 번째 단계에서는 오타 주소에 대한 오류 정제를 수행한다. 주로 한자어표기에 따른 오류(예 : 쌍용리 -> 쌍룡리)나 주소 입력 시 생길 수 있는 오타(서울->사울) 등을 처리한다. 오타 주소의 정제방법은 <그림 6>과 같이 시도 표준어 형태와 시도, 시/군/구, 읍/면/동, 리, 도서의 단어를 기준으로 입력된 주소를 분리하고 다시 자소단위로 분리하여 단위 주소 간의 유사도를 비교한다. 이 때 단위 주소간의 유사도는(식 1)과 같이 초성, 중성, 종성 별 일치 개수를 전체 글자 수로 나눈 값을 사용하여 계산한다.



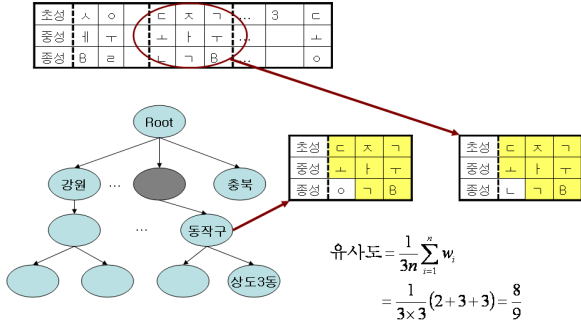


그림 6. 오타 주소 정제

이렇게 구해진 유사도는 미리 정해진 임계값을 만족할 때 선택된다. 임계값의 기준은 최대한 허용할 수 있는 오타의 범위를 의미하며 <표 7>을 참조하여 정하였다.

표 7. 주소 오타 허용 범위

	서울(6)	동작구(9)	남양주시(12)	노량진1동(15)
1자소	0.83(5)	0.88(8)	0.92(11)	0.93(15)
2자소	0.66(4)	0.77(7)	0.83(10)	0.86(13)
3자소	0.50(3)	0.66(6)	0.75(9)	0.80(12)
4자소	0.30(2)	0.55(5)	0.66(8)	0.73(11)

<표 7>에서 열의 괄호 안 숫자는 단위 주소의 자소의 총 개수를 의미하고 행은 몇 개의 자소에서 오타가 발생했는지를 보여준다. 임계값이 너무 낮을 경우에는 오타에 대한 허용기준이 느슨하여 검색의 효율성이 떨어지고, 임계값이 너무 높을 경우에는 오타의 허용기준이 엄격해 검색이 되지 않는 경우가 있기 때문에 본 시스템에서는 임계값을 1, 2자소의 오타만 처리하도록 0.83으로 설정하였으며 <그림 7>과 같이 임계값 이상을 만족하는 항목의 값을 선택하여 주소를 정제한다

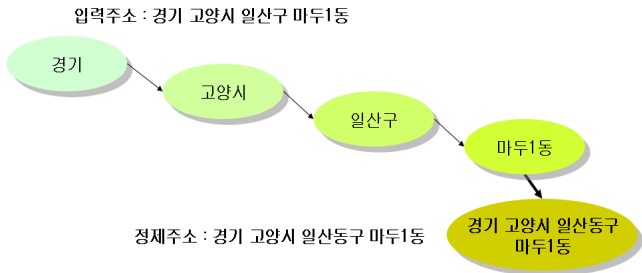
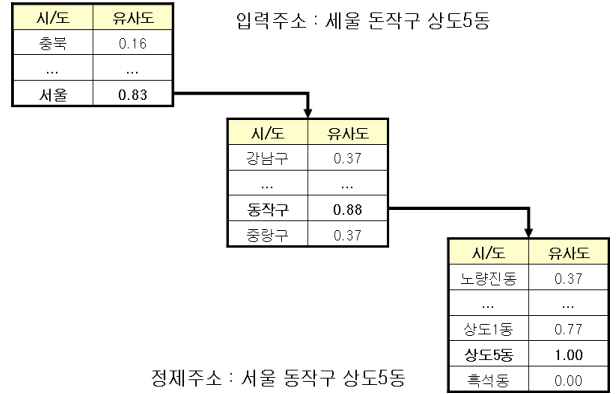


그림 7. 구주소 트리

네 번째와 다섯 번째 단계에서는 행정구역 변경 등으로 주소가 변경되었는데 변경되기 이전의 구주소가 입력되었을 때 이를 신주소로 변경하는 단계이다 예를 들어서 "경기도 고양시 일산구의 경우 2005년 5월 일산동구와 일산서구로 분리되었다 따라서 "경기 고양시 일산구 마두1동"과 같은 주소는 "경기 고양시 일산동구 마두1동"으로 정제되어야 한다. 이 단계에서는 <그림 8>과 같은 구주소 트리를 이용하여 구주소를 정제 하며 생성된 구주소 트리의 단말노드에는 새롭게 변경된 주소가 위치한다. 구 주소 변경을 이용하여 신 주소체계가 확립되는 2008년 이후에는 효율적으로 구 주소를 신 주소로 수정 할 수 있다.



정제주소 : 서울 동작구 상도5동

그림 8. 한글 오타 처리

여섯 번째, 일곱 번째, 여덟 번째 단계는 사용자가 모든 주소 정보를 입력하지 않고 특정한 정보만 입력하였을 때 수행되는 과정이다 이 단계에서는 읍/면/동의 이름을 키(key)로, 해당 읍/면/동 이름의 상위 주소를 값(value)으로 지정하여 구성한 역 주소테이블을 이용한다 단, 이 경우에는 읍/면/동의 이름이 유일한 경우에만 정제된다. 이 단계에서 이용하는 역 주소테이블의 예는 <표 8>과 같다.

표 8. 역 주소테이블의 예

키	값
강동면	강원 강릉시
용호동	경남 창원시
	경북 김천시
	대전 대덕구
	부산 남구
...	...

마지막 아홉 번째 단계에서는 주소트리에서 단말노드까지 검색이 되지 않았거나 두 개 이상의 주소가 검색되는 경우를 처리한다 만약 법정동 정보만 이용하여 주소에 우편번호를 부여해야할 경우에는 법정동 정보만을 이용하여 우편번호를 부여하게 된다. 또, 두 개 이상의 주소가 검색되는 경우에는 주소 정제를 실패(reject)하게 된다. 예를 들어 원시 주소가 "연수동 유원1차 아파트"인 경우 연수동이 "인천 연수구 연수동"과 "충북 충주시 연수동"처럼 두 개의 동이 존재하므로 정제를 실패하게 된다

#### 4. 실험 결과

##### 4.1 실험 환경

본 논문에서 제안한 주소 정제 시스템인 AddressClean은 다양한 어플리케이션과의 효율적인 통합을 위하여 Java 플랫폼에서 개발되었으며 실험에 이용된 시스템은 Pentium(R) 4 1.7GHz, 512MB 메모리, 운영체제는 Microsoft Windows XP Professional SP2를 사용하였다.

##### 4.2 수행 속도 및 정확도 평가

AddressClean의 정확도와 수행 속도를 측정하기 위하여 실제로 사용되는 주소 데이터 3,394건과 주소에 오류가 존재하지 않는 10만건의 데이터를 구축하였다. 실제 데이터는 표준화 되지 않은 데이터이므로 오류율을 예측하기 위하여 전체 데이터 3,394건에서 200건의 데

이터를 샘플링하여 사용하였으며 10만건의 데이터는 오류가 존재하지 않는 200건의 데이터를 이용하여 만든 실험 데이터이다. <표 9>의 실험 결과에서 볼 수 있듯이 3,294건의 실제 데이터 중 27건만 정제에 실패 하였다. 전체 데이터의 오류율을 알아보기 위해 200개를 샘플링하여 오류 주소를 검사한 결과 잘못 정제된 5개의 주소를 발견하였다. 이를 바탕으로 전체 정제 오류를 계산한 결과 3,394개의 데이터에서 약 2.5%인 85개 정도의 주소가 잘 못 정제될 것으로 예측 되었다. 오류가 없는 10만건의 데이터를 정제한 결과 정제에 실패한 경우는 없었으며, 전체 소요시간은 376초로 1건당 약 4ms의 시간이 소요되었다.

표 9. 실험 결과

실제 데이터 (3,394건)	정제 건수	정제 성공 : 3,367건 (99.2%) 정제 실패 : 27건 ( 0.8%)
	정제 소요시간	1건당 평균 : 4.35ms 전체 시간 : 14,776.09ms
	오류율	200건내 오류 : 5/200 (2.5%) 예상 오류 개수 : 85/3394
실험 데이터 (10만건)	정제 건수	정제 성공 : 100,000건 (100%) 정제 실패 : 0건 ( 0%)
	정제 소요시간	1건당 평균 : 3.76ms 전체 시간 : 376,490.22ms

4.3 성능 비교

기존의 다른 주소 정제 시스템들과의 비교를 위하여 우편번호가 있는 80개의 데이터를 이용하여 오타와 누락 띄어쓰기 오류가 있는 데이터를 각각 80건씩 생성하였다. 이러한 데이터를 AddressClean과 S업체의 주소정제 시스템, E업체의 주소정제 시스템으로 각각 주소 정제를 수행하였으며 오류율에 대한 비교 결과는 <표 10>과 같다.

표 10. 기존 시스템과의 오류율 비교

	AddressClean			S업체 시스템			E업체 시스템		
	오류	정제 실패	오류율	오류	정제 실패	오류율	오류	정제 실패	오류율
띄어쓰기	0	0	0.00	1	1	1.27	5	2	6.41
오타	1	7	1.37	2	17	3.17	4	59	19.05
누락	0	4	0.00	2	8	2.78	7	23	12.28
합계	1	11	0.43	5	26	2.24	16	84	10.26

비교 결과 본 논문에서 제안하는 AddressClean은 다른 주소정제 시스템에 비하여 띄어쓰기 오타, 누락 주소 모두 낮은 오류율을 보였으며, 특히 정제 실패 비율도 4.5%로 기존의 주소 정제 시스템인 S업체(10.8%)와 E업체(35%)에 비하여 우수함을 확인하였다. <표 10>에서 오류는 정제를 수행 했으나 틀린 경우이고 정제 실패는 해당 정제 시스템에서 정제를 할 수 없는 데이터를 분류하여 정제하지 않은 데이터를 말한다 오류율은 실질적으로 정제 과정을 거친 주소 중 오류가 발생한 경우만을 평가한 것으로 (식 2)와 같이 계산하였다.

$$\text{오류율} = \left( \frac{\text{정제결과가틀린 주소의 수}}{\text{전체주소수} - \text{정제실패수}} \right) * 100 \quad (\text{식 } 2)$$

본 논문에서 제안하는 AddressClean과 기존의 다른 시스템들의 초당 처리 속도를 비교한 결과는 <표 11>과 같다. AddressClean은 어셈블리어로 개발되어 처리 속도가 빠른 S업체의 주소 정제 시스템 보다는 느리지만 AddressClean과 같이 일반 고급 프로그래밍 개발 언어로 개발된 O업체와 G업체의 주소 정제 시스템 보다 빠른 처리 속도를 보인다 S업체의 시스템은 어셈블리어로 개발하여 유지보수와 프로그램 업데이트가 매우 어렵지만 AddressClean은 Java 플랫폼에서 개발되어 유지보수와 프로그램 업데이트가 S업체의 시스템에 비해 훨씬 수월하다

표 11. 기존 시스템과의 초당 처리 속도 비교

주소정제 시스템	초당 처리 속도	비고
AddressClean	266건	
S업체	약 2,000건	어셈블리어로 개발됨
O업체	43건	
G업체	167건	

5. 결 론

본 논문에서는 효율적인 주소정제 시스템인 AddressClean을 개발하였다. AddressClean은 일관성 없는 주소를 표준 형태로 변환하기 위하여 주소 트리 생성을 통한 정확한 주소 검색 유니코드 분석을 이용한 한글 입력 실수 정제, 역 주소 테이블을 사용한 누락 주소 정제, 한글 유사도 비교를 이용한 오타 주소 정제 방법들을 제안하여 개발하였다. 실험 결과를 통하여 정확한 주소에 대해서는 초당 약 266건의 정제가 가능하고 80개의 오타와 누락, 띄어쓰기 오류 등 정확하지 않은 주소에 대해서는 이미 제안된 타 정제 시스템 보다 정제율은 높고 오류율은 적음을 확인함으로써 제안한 정제 방법들의 타당성을 검증하였다. 따라서 AddressClean은 다른 주소정제 시스템에 비하여 주소정제가 정확하고 빠르며 구주소 변환 모듈을 이용하여 주소 체계가 변경 된 후에도 사용이 용이하다. 따라서 일관성 없고 부정확한 주소를 정제함으로써 잘못된 주소로 DM을 발송할 때 생기는 비용을 절약할 수 있으며, 개인화 정보 서비스의 질적 향상을 위하여 활용이 가능하다. 또한, 고객의 지역별 성향 거주 지역의 환경에 관한 정보와 교통 및 서비스에 관련된 정보를 이용하여 차별화된 타겟 마케팅에 활용 할 수 있다

6. 참고 문헌

- [1] 정보통신부, 우편번호 안내 <http://www.koreapost.go.kr>
- [2] 수지원넷 소프트, <http://www.sujiewon.co.kr>
- [3] 이스턴 테크놀로지, <http://www.easterntech.co.kr>
- [4] 공영DBM, <http://www.00DB.co.kr>
- [5] OPENmate, <http://www.openmate.co.kr>
- [6] The Unicode Standard, Version 4.0, <http://www.unicode.org>