

# POI 이형태 데이터베이스 구축 시스템

## (A Point-Of-Interest Allomorph Database Construction System)

양 승 원 <sup>†</sup>      이 현 영 <sup>\*\*</sup>      왕 지 현 <sup>\*\*\*</sup>  
 (Seung Weon Yang) (Hyun Young Lee) (Ji Hyun Wang)

**요 약** 내비게이션 시스템에서 목적지를 찾기 위하여 목적지의 이름, 분류, 주소, 전화번호 등의 정보를 이용하게 되는데 대부분의 사용자들은 이들 중에서 목적지의 이름을 사용한다. 그런데 사용자들은 공급사에서 제공하는 POI DB에 등재된 이름을 정확히 알지 못할 뿐만 아니라 편의상 축약된 명칭이나 일반적으로 불리어지는 명칭 등으로 POI 검색을 시도하므로 검색이 실패하는 경우가 빈번하다. 본 논문에서는 내비게이션 시스템에서 이름으로 검색 할 때 검색 성공률을 제고할 수 있는 이형태 DB 구축 시스템을 제안한다. 이 시스템은 원 DB의 POI 명과 연결되는 이형태를 생성하여 DB화한다. 우리는 이형태의 생성을 위하여 약 650,000 개의 개체를 가지고 있는 원 DB의 POI 명으로부터 모은 패턴을 분석하여 이형태의 유형을 7 가지로 분류하였다. 분류한 유형을 토대로 일정한 패턴이 존재하여 자동화가 가능한 유형들에 대하여 577개의 규칙을 만들어 자동으로 이형태를 구축하였다. 규칙으로 만들기가 어렵거나 빈도수가 적은 개체들에 대해서는 수동으로 이형태를 구축하였다. 생성된 비율은 전체 POI DB의 35.8%에 해당하며 구축한 이형태 DB를 사용한 검색 성공률은 89%이었다.

**키워드** : Point-Of-Interest(POI), POI 이형태, 이형태 DB

**Abstract** People use various information for searching POI in the navigation system such as name, category, address, phone number. Most of users use name and category to search their POI. They don't know exact name in POI DB provided by Maker. They use abbreviated or generalized name as key word for searching POI. Because of these reasons, the hit ratio has been very low.

In this paper, We suggest a extra DB\_construction system for raising the hit ratio. It generates allomorphes DB link to the POI name in original DB. We classified the POI names in original DB into seven types of allomorph by analyzing the gathered patterns from the POI DB which has over 650,000 entries. For auto\_generating the allomorphes, we made 577 rules based on the classified types. And we generated the allomorphes manually for the entries which are difficult to make the rule and has low frequency. The generated allomorphes account for 35.8% of all original DB. The hit ratio is 89% under suggested system.

**Key words** : Point-Of-Interest(POI), POI Allomorph, Allomorph DB

### 1. 서론

대부분의 차량에 내비게이션이 장착될 정도로 내비게이션이 일반화되고 있는 현 시점에서, 여행자가 목적지(POI:관심지점)를 찾아가기 위해 내비게이션에 의존하는 비중은 갈수록 높아지고 있다. 내비게이션의 사용자 관점에서 볼 때 목적지를 보다 효율적으로 검색할 수 있도록 해 주는 것이 가장 중요한 기능들 중의 하나이다. 이러한 필요성에 따라 지금까지 목적지 검색에 관한 연구가 진행되었다[1-3]. 이러한 연구들은 POI 데이터베이스를 구축하는 데에 주된 관심을 가지고 있었고 본 논문에서 제안하는 검색의 정확도나 편의성을 제고하는 점과는 거리가 있다. 또 데이터베이스의 엔트리로부터

· 이 논문은 한국전자통신연구원과 우석대학교의 지원에 의해 이루어졌음

<sup>†</sup> 중신회원 : 우석대학교 게임콘텐츠학과 교수  
 yangy123@ws.ac.kr

<sup>\*\*</sup> 학생회원 : 전북대학교 컴퓨터학과  
 lhy123@hanmail.net

<sup>\*\*\*</sup> 정회원 : 한국전자통신연구원 음성언어설 선임연구원  
 jhwang@etri.re.kr

논문접수 : 2008년 8월 8일

심사완료 : 2008년 12월 23일

Copyright©2009 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 소프트웨어 및 응용 제36권 제3호(2009.3)

약칭어를 자동으로 추출하여 웹문서의 검색 성공률을 올리려는 시도도 있었다[4,5]. 이 연구들에서는 통계적인 방법을 이용하여 그리드형태로 후보(candidate)들을 생성하였으나, 후보가 너무 많아질 우려가 있을 뿐만 아니라 결과가 학습에 좌우되고 약칭에 한정되어 있어 내비게이션 POI 데이터베이스에 적용하기 힘들다.

내비게이션은 POI(관심지점)를 검색하기 위해서 다양한 방법을 제공하지만 주로 이용되는 기능은 POI명이나 분류 검색과 소요시간 검색, 길안내 요청, 전화번호 및 주소 검색 등이다. 사용자들은 POI명을 이용한 검색 서비스를 가장 빈번하게 이용하며, 이 때 사용자는 보다 편리하게 원하는 지점을 찾기 위해 그들이 알고 있는 다양한 명칭을 키워드로 사용한다. 그러나 현재 사용 중인 대부분의 내비게이션은 POI명으로 검색할 때 POI 데이터베이스에 등록된 POI명과 사용자가 입력한 키워드를 비교하여 찾아주는 방법을 이용하기 때문에 데이터베이스에 등록된 POI 명을 정확히 입력하지 않으면 목적지를 찾아주지 못하게 된다. 또한 대부분의 사용자들은 데이터베이스에 등록된 POI명을 정확히 알지 못하는 경우가 많고 명칭을 줄여 쓰는 경향이 있어 내비게이션의 검색 성공률은 매우 낮을 수밖에 없다.

예를 들어, POI 명칭이 “한국표준과학연구원”인 경우에 사용자는 데이터베이스에 등록된 POI명을 알지 못하거나 편의상의 이유로 “표준과학연구원”, “표준연”, 혹은 영어식 약자 표기인 “kriss” 및 “크리스”로 검색을 시도할 수 있다. 그러나 현재 사용 중인 대부분의 내비게이션의 데이터베이스에는 POI 명칭으로 “한국표준과학연

구원”만이 등록되어 있으며 검색알고리즘은 전방일치 방법을 사용하기 때문에 이들 키워드를 이용하여 검색을 시도하는 사용자의 검색 요구를 정확히 충족시킬 수 없게 된다.

그림 1과 그림 2는 널리 사용되고 있는 두 개의 제품 [4,5]을 대상으로 “한국표준과학연구원”을 검색하기 위해 “한국표준”, “표준연” 그리고 “표준과학연구원”을 검색어로 하여 검색한 결과이다. 위의 그림들에서 볼 수 있듯이 검색어를 “표준과학연구원”과 “표준연”으로 검색할 경우에는 두 개의 제품 모두에서 원래 의도한 “한국표준과학연구원”을 찾을 수가 없다. “한국표준”으로 검색할 경우에는 “한국표준”을 머리어로 갖는 엔트리들이 후보 POI로 검색되었다. 이는 POI 데이터베이스에 “한국표준과학연구원”만 등록되어 있고 전방일치 검색을 사용하는 데에서 기인한 것이다. 유사한 예로, POI명이 “OO초교”로 등록되어 있다면 “OO초등학교”로 검색하면 사용자가 원하는 POI를 찾을 수 없을 뿐만 아니라 반대로 POI 명이 “OO초등학교”라면 “OO초교”로 검색할 경우 목적하는 POI를 찾을 수 없게 된다.

이러한 문제를 해결하기 위해서 본 논문에서는 POI명에 대한 이형태를 생성하여 원래 POI 데이터베이스(원 DB)의 POI명과 매핑하는 시스템을 제안한다. 이형태를 생성하기 위하여 POI 데이터베이스에 있는 POI명의 패턴을 찾아 생성 가능한 이형태 유형을 분류하고 자동화가 가능한 규칙을 찾아낸다. 그런 다음 규칙들에 여러 제약 조건 등을 추가하여 이형태 자동 생성 규칙을 완성한다. 마지막으로 생성된 규칙을 이형태 생성 시

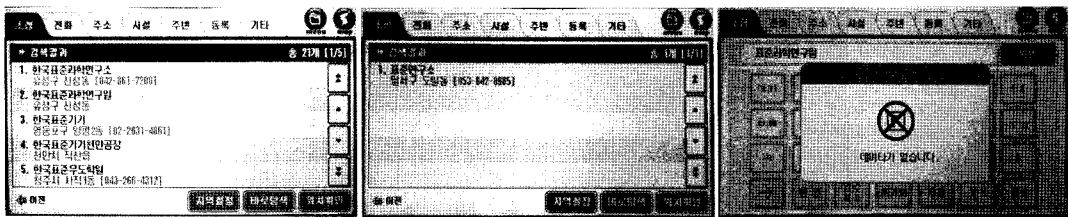


그림 1 A사 제품의 검색 결과

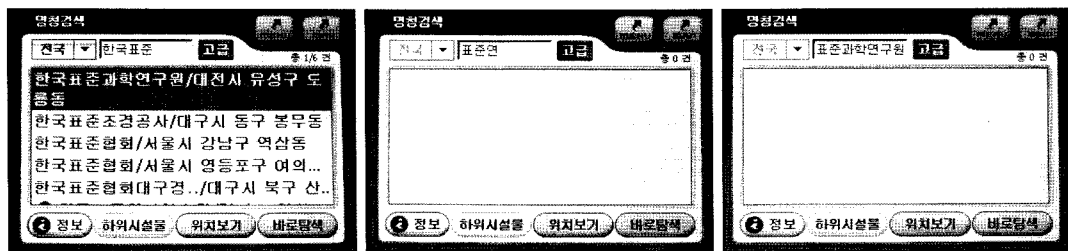


그림 2 B사 제품의 검색 결과

시스템을 이용하여 POI명에 대한 이형태를 자동으로 생성하고 생성된 이형태는 원 DB와의 매핑에 이용한다. 이렇게 생성된 POI명의 이형태는 다양한 키워드로의 검색을 가능하게 해 줄 뿐만 아니라 검색 성능도 향상됨을 보인다.

## 2. POI의 이형태 유형

내비게이션 사용자는 원하는 지점(POI)을 검색할 때, 검색의 편의성 및 습관적 사용을 이유로 핵심 의미어 위주로 검색하는 경향이 있다. 본 논문에서는 원 DB에 등록되어 있지 않지만 보통의 사용자가 검색어로 사용할 가능성이 매우 높은 의미를 POI명에 대한 이형태라고 한다. 새로 정의된 이형태는 검색 단계에서 원 DB의 POI명에 매핑되어 검색의 적중률을 높인다. 본 논문에서는 POI명에 대한 패턴을 분석하여 이형태의 유형을 생략형, 축약형, 혼용형, 약칭형, 대표형, 별칭형, 이동형의 7가지로 분류하여 사용하였다.

### 2.1 생략형

지역 명이 붙어 4글자 이상으로 형성된 POI 명칭에서 부가된 지역 명을 생략하는 형태와 6글자 이상의 국가기관이나 공기업의 명칭에서 국가 명을 생략하는 형태, 그리고 아파트나 방송국과 같은 대분류 항목이 POI에 중복 등장할 때 대분류 명칭을 생략하는 유형이다.

#### 2.1.1 지역명 + 기관명

대전가수원동우체국 ↔ 가수원동우체국

군산산업단지 ↔ 산업단지

예외 : 대구탕(일식전문점) != 탕

군산항(항만,항구) != 항

#### 2.1.2 “한국” + 관공서/연구소

한국전자통신연구원 ↔ 전자통신연구원

한국농촌공사 ↔ 농촌공사

예외 : 한국전력 != 전력

한국통신 != 통신

#### 2.1.3 대분류 생략

신창코아루아파트 ↔ 신창코아루

경산태성아파트 ↔ 경산태성

### 2.2 축약형

부분적으로 축약되어 사용되는 형태로 “초/중/고/대학(교)”와 같은 교육기관에서 “학(교)”를 단축하는 형태와 공원에 붙는 국립/도립/시립/군립 등을 삭제하거나 대교나 철교 등의 교량에서 대/철을 삭제하고 교로 축약하는 유형과 읍/면/동/시/군/구와 같이 행정구역명이 포함되었을 때 이를 생략하는 유형이다.

#### 2.2.1 학교 명칭 형태

가경초등학교 ↔ 가경초교

고창북고등학교 ↔ 고창북고, 고창북고교

거진여자중학교 ↔ 거진여자중, 거진여중, 거진여중학교 (“여자”, “실업”, “상업”, “공업”, “외국어”, ... 등 포함)

#### 2.2.2 국립/도립/시립/군립 공원 형태

대둔산도립공원 ↔ 대둔산공원

변산국립공원 ↔ 변산공원

#### 2.2.3 대교/철교 등의 교량 명칭 형태

영동대교 ↔ 영동교

한강철교 ↔ 한강교

#### 2.2.4 행정구역명이 포함된 명칭에서 시/군/구/읍/면/동 생략

평화동제일아파트 ↔ 평화제일아파트

소양면우체국 ↔ 소양우체국

완주군보건소 ↔ 완주보건소

### 2.3 혼용형

명칭이 변경되거나 영어의 한국식 표현 등으로 인해 일반인이 혼용하여 사용하는 형태와 하위 개념을 상위 개념으로 혼용하여 사용하는 유형이다.

#### 2.3.1 명칭이 변경된 형태

한국전자통신연구소 ↔ 한국전자통신연구원

국민학교 ↔ 초등학교

동사무소 ↔ 주민자치센터

#### 2.3.2 영어의 한국식 표현 형태

모텔 ↔ 여관

빌라 ↔ 연립

#### 2.3.3 영어 발음상 혼용

렌터카 ↔ 렌트카

슈퍼/마켓 ↔ 수퍼/마켓

데이터 ↔ 데이타

샤시 ↔ 샷시, 사슈, 샷수

센터 ↔ 쎄터, 쎄타, 쎄타

#### 2.3.4 상하위 개념이 명확한 경우 하위 개념을 상위 개념으로 혼용

보건소 ← 보건지소, 보건분소, 보건진료소, 보건소분소, 보건센터

초분교/중분교/고교분교 ← 초/중/고교

#### 2.3.5 일반적인 혼용

전시관 ↔ 전시장

농장 ↔ 농원

수련관 ↔ 수련원

식물관 ↔ 식물원

#### 2.3.6 한국어와 영어의 혼용

미니스톱 ↔ ministop

빕스 ↔ vips

### 2.4 약칭형

국가 기관이나 국영 기업에서 편의성을 위해 축약해 부르던 것이 대학, 기업, 연구소 등으로 확대 사용되는

형태로 대부분 단어의 앞 글자를 따서 약칭형으로 분류하나 예외인 경우도 있다.

2.4.1 국가 기관의 부/원으로 끝나는 경우 단어마다의 머리글자와 부/원을 붙임

지식경제부 ↔ 지경부

교육과학기술부 ↔ 교과부

2.4.2 검찰 관련 부서의 경우 단어의 머리글자만 붙임

지방법원 ↔ 지법

고등법원 ↔ 고법

지방검찰청 ↔ 지검

예외) 대검찰청 -> 대검, 대법원 -> 대법

2.4.3 기타(규칙이 적용되지 않는 형태)

포항제철 ↔ 포철

식품의약품안전청 ↔ 식약청

**2.5 대표형**

주로 전국에 몇 안 되는 유일한 명칭이나 시/군/구/도 등에서 하나만 존재하는 “지명+명칭” 형태의 POI인 경우 지명을 생략해도 구별이 가능한 유형이다.

금정경륜장 ↔ 경륜장

전주KBS방송국 ↔ KBS방송국, KBS

**2.6 별칭형**

관공서나 연구기관, 대기업, 공공기업, 신문사, 방송국, 은행 등은 기관명 대신 영문 명칭을 사용하는 추세인데, 이러한 명칭들을 모아 분류한 유형이다.

한국전자통신연구원 ↔ ETRI

한국통신 ↔ KT

한국방송공사 ↔ KBS

기업은행 ↔ IBK

**2.7 이동형**

명칭 중의 일부 순서가 뒤바뀌는 형태로 호텔이나 모텔과 같은 경우에 주로 나타나는 유형이다.

호텔신라 ↔ 신라호텔

모텔가나 ↔ 가나모텔

위와 같이 POI명에서 생성 가능한 이형태의 유형을 7가지로 분류하고 이 중에서 자동화가 가능한 생략형, 축약형, 혼용형, 대표형, 이동형은 본 논문에서 제안한 규칙을 기술하여 자동으로 이형태를 생성하고, 특별한 규칙이 존재하지 않거나 규칙으로 기술하기에는 유형이 너무 적은 약칭형, 별칭형 등의 이형태는 수동 도구를 이용하여 수작업으로 추가한다. 약칭형이나 별칭형 중에서도 POI 데이터베이스에서 출현 빈도가 높은 경우는

자동으로 생성하기 위한 규칙을 추가하여 이형태를 자동 생성한다.

**3. 이형태 자동 생성**

본 장에서는 POI명으로부터 이형태를 자동으로 생성하기 위해 POI의 이형태 유형 중에서 일정한 패턴이 존재하는 생략형, 축약형, 혼용형, 대표형, 이동형 등의 패턴을 규칙으로 기술하고 이 규칙으로부터 이형태를 생성하는 방법을 기술한다.

**3.1 이형태 생성 규칙**

POI 명에 대한 이형태를 자동으로 생성하기 위해 사용되는 규칙은 그림 3과 같이 3개의 부분으로 구성된다.

Rule ID	Rule Description	Rule Body
---------	------------------	-----------

그림 3 POI 생성 규칙의 구조

Rule ID는 이미 적용된 규칙을 삭제할 경우 삭제 정보로 활용되며 시스템에 의해서 자동으로 생성되는 규칙 번호이다. 또한 Rule Description은 규칙에 대한 설명(Comment)을 작성하는 곳으로 규칙의 중복 작성을 방지하거나 관리하기 위한 정보로 활용한다. Rule Body에는 Rule Type 및 여러 가지 제약 조건 등을 포함한 실질적인 규칙을 작성하는 부분으로 POI 명에 대한 이형태를 생성하는 정보로 활용한다. 이러한 구조를 바탕으로 작성한 규칙의 예를 보이면 그림 4와 같다.

그림 4에서 [1199285521]이 Rule ID에 해당된다. Rule Description에서 #589는 사용자가 룰을 관리하기 위해 지정한 룰 번호이며 뒤따르는 부분은 룰에 대한 상세 설명 부분이다. Rule Body는 [!+초등학교][TE((%SUB%\*분교)(%TRA%초교분교|초등학교분교|초분교|초등학교|초교))]이다.

POI 명의 이형태 자동 생성 규칙의 Body는 [대분류 제약조건][규칙]으로 기술된다. [규칙]은 규칙의 시작을 알리는 구분자 T를 시작으로 6가지 타입(C/D/R/X/M/I)이 있으며 주제어를 중심으로 기술된다. 주제어는 원 DB의 POI명 필드의 내용 중 규칙을 적용할 부분을 말한다. 규칙의 타입에 사용되는 C(Cross)는 “논합”은 “농업협동조합”으로 “농업협동조합”은 “농협”으로 변환할 때와 같이 주제어와 특정 단어를 서로 이형태로 생성할 때 사용한다. D(Delete)는 특정 단어를 삭제할 때, 예를 들면 “서울대학교”에서 “학교”를 삭제할 때 사용한

1199285521	[#589-초등학교-분교로 끝나면 초교분교 초등학교분교 초분교 초등학교 초교]	[!+초등학교][TE((%SUB%*분교)(%TRA%초교분교 초등학교분교 초분교 초등학교 초교))]
------------	---	--

그림 4 반자동 생성 룰에 의해 생성된 규칙의 예

다. R(Replace)은 “농원”을 “농장”으로 바꿀 때와 같이 주제어를 특정 단어로 대체할 때 사용하며 POI의 유형이 혼용형일 때 주로 사용되는 타입이다. X(exclusive)는 주제어의 앞이나 뒤의 단어를 제거할 때 이용한다. 예를 들면 “XX보건소YY지소”의 경우 “보건소”뒤에 나오는 단어를 모두 제거하고 “XX보건소”라는 이형태를 만들 때 사용한다. M(Move)은 “모텔가나”를 “가나모텔”로 생성되도록 주제어와 특정 단어의 위치를 이동할 때 사용하며 I(Insert)는 주제어의 앞이나 뒤에 새로운 단어를 추가할 때 사용한다. 이들 타입을 포함하는 Rule Body 부분을 간략히 표현하면 그림 5와 같다.

```
Tt[{{Rule_Words}}]+
t ::= C | D | R | X | M | I
Rule_Words ::= RuleW|,RuleW)*
RuleW ::= (((길이제약))([단어제약]))%주제어%([길이제약]))([단어제약])),
길이제약 ::= Operator Length
Operator ::= ">" | "<" | "<=" | ">=" | "=" | "!="
Length ::= Number
단어제약 ::= (+|-)단어[|단어]*
+ : Exist
- : Not Exist
```

그림 5 이형태 생성 규칙의 Rule Body

이형태를 생성하기 위해서는 다양한 조건이 반영되어야 한다. 이런 조건은 규칙에서 제약 조건으로 표기되는데 본 시스템에서는 “대분류 제약”, “단어 제약”, “길이 제약”을 설정할 수 있다. “대분류 제약”은 주제어와 무관하게 특정 대분류로 규칙의 적용 범위를 한정할 때 사용하며 주제어의 앞이나 뒤에 올 수 있는 특정 단어에 대한 제약에는 “단어 제약”을 사용하고 POI 문자열에 대한 길이를 제약할 때는 “길이 제약”을 사용한다. 그림 6은 이들 제약을 적용하기 위해 규칙으로 기술하기 위한 기호를 나타내고 있다.

```
'*' : length >= 0
':' : 금지어
'+' : 허용, 기본 값
'|' : OR
'<', '>': 방향
'$' : 주제어포함여부
```

그림 6 규칙에 사용되는 제약 기호

작성된 규칙은 룰 해석기가 해석하여 POI 데이터베이스의 모든 엔트리에 대해 해당 규칙에 대한 이형태를 생성한다.

### 3.2 룰 편집기

앞 절에서 정의한 규칙은 그 구조가 제법 복잡하며 기술해야할 규칙의 개수가 매우 많은 것을 감안하면 규

칙을 기술하는 데에 투자해야 하는 시간이 너무 부담스럽고 규칙의 일관성 또한 기대하기 힘들다. 따라서, 본 연구에서는 규칙을 쉽게 작성하고 수정할 수 있도록 룰 편집기를 구현하여 사용하였다.

예를 들어, “초등학교”라는 대분류 항목(Category)에서 “분교”로 끝나는 경우 “초교분교|초등학교분교|초분교|초등학교|초교”라는 이형태를 룰 편집기를 이용하여 생성해 보자. 이를 수행하기 위한 순서도는 그림 7, 그리고 규칙 편집기는 그림 8과 같으며 설명의 편의상 그림 안에 번호를 붙여 두었다. 생성된 규칙을 일관성있게 관리하기 위하여 “Rule Description(1)”, “Type(5)”, “Subject(4)”는 반드시 기술하여야 하며 나머지 필드에는 생성하고자 하는 규칙에 따라 적절하게 제약 조건을 기술하면 된다. 제약 조건을 모두 기술한 후에 추가(+) 버튼을 누른 다음 “Generation”을 실행하면 규칙이 생성된다. “Rule Add” 버튼은 생성된 규칙을 규칙 DB에 저장하는 최종 확인버튼이다. 과정 (10)은 작성된 룰이 제대로 수행되는지를 확인해보기 위해 필요한 단계로 생략 가능하다.

기존의 규칙을 편집하기 위해서는 해당 규칙을 더블 클릭하면 이미 입력된 규칙 정보가 룰 편집기 상태에 반영이 되어 나타나므로 해당 규칙의 편집이 가능하게 된다. 한 항목에 여러 규칙을 추가할 경우에는 위의 (4)~(7)의 과정을 반복 수행하여 규칙들을 추가하면 되며 대분류 항목이나 규칙 type이 같은 경우에만 가능하다. 이렇게 하여 생성된 규칙을 몇 가지 나열하면 다음과 같다.

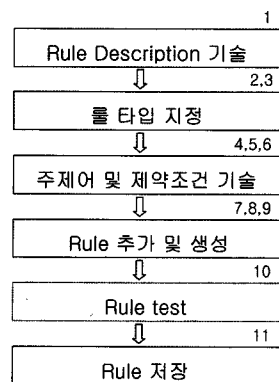


그림 7 Rule 생성 과정

## 4. POI 이형태 데이터베이스 구축 시스템

이번 절에서는 앞에서 설명한 모듈들을 통합하여 이형태 데이터베이스를 구축한 이형태 데이터베이스 구축 시스템에 대하여 설명한다. 통합된 시스템의 구성도는 그림 10과 같다. 통합시스템에서 룰 편집기는 POI의 이

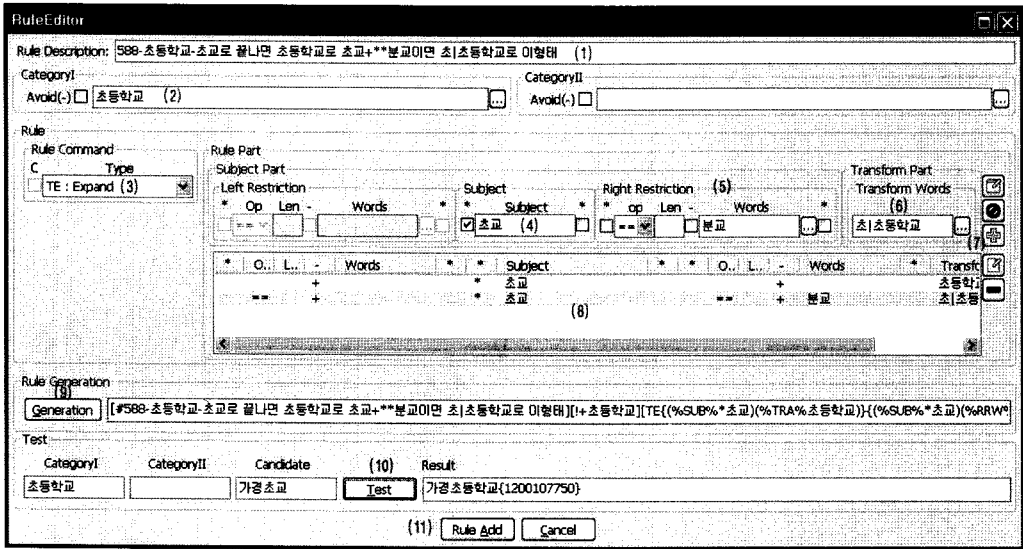


그림 8 Rule Editor

```

1199285521 [#589-초등학교-분교로 끝나면 초교분교|초등학교분교|초분교|초등학교|
초교] [!+초등학교][TE((%SUB%*분교)(%TRA%초교분교|초등학교분교|초분교|
초등학교|초교))]

1197467898 [#355-대학교-대로 끝나고 바로 앞에 의교|예|여|과|외|공이 오지 않으
면 대학|대학교로변경] [!+대학교][TE((%RLW%-의교|예|여|과|외|공)(%LLS%
(%SUB%대) (%TRA%대학|대학교))]

1197414169 [#345-농/축협-농협<->농업협동조합, 축협<->축산업협동조합, 수협
<->수산업협동조합으로변경] [!+농/축협][TC((%SUB%*농협|농업협동조합*)
((%SUB%*축협|축산업협동조합*))((%SUB%*수협|수산업협동조합*))]
    
```

그림 9 자동 생성을 위해 작성된 규칙

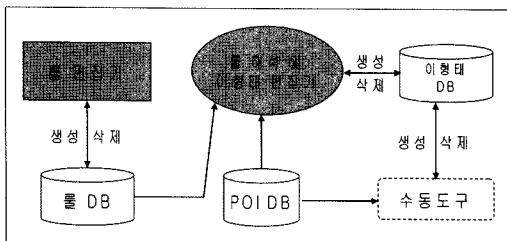


그림 10 통합 시스템의 구성도

형태를 생성하기 위한 규칙들을 작성하거나 삭제할 수 있는 모듈이며 생성된 물은 물 DB에 저장한다. 물 해석 및 이형태 편집기는 물 DB를 참조하여 생성된 물에 대한 해석을 수행하고 이를 근거로 해당 물에 대한 이형태를 생성한다. 생성된 이형태는 이형태 DB에 저장되며 검색할 때 검색 정보로 활용되어 POI DB와 매핑된다. 따라서 본 논문에서 제안한 시스템 하에서 POI를 검색

할 때 POI DB에 있는 POI명만 이용하여 검색하는 것이 아니라 이형태 DB에 있는 이형태들이 우선 검색 대상이 되어 사용자가 다양한 형태의 POI 명을 입력하더라도 원하는 POI명을 검색할 수 있다. 또 수동도구는 규칙으로 작성되지 않은 POI명에 대한 이형태를 생성하거나 수정할 때 사용된다.

본 논문에서 구현한 이형태 데이터베이스 구축시스템의 실행 화면은 그림 11과 같다. 왼쪽 상단 부분이 대부분류 항목을 보여주며 오른쪽 상단은 선택된 대부분류에 속하는 POI 명을 보여준다. Allomorph는 수작업으로 이형태를 생성하는 필드이며 Auto 필드는 규칙을 해석하여 자동으로 생성된 이형태를 보여준다. 그림 11의 예에서는 대부분류 항목이 초등학교(5003 항목을 가짐)에서 체크(V) 표시된 588번 물을 실행한 결과 4948개의 항목에 적용되었음을 보여주고 있다. 전체 항목수(5003)와 규칙이 적용된 항목수(4948)과는 차이가 있는데 그 이

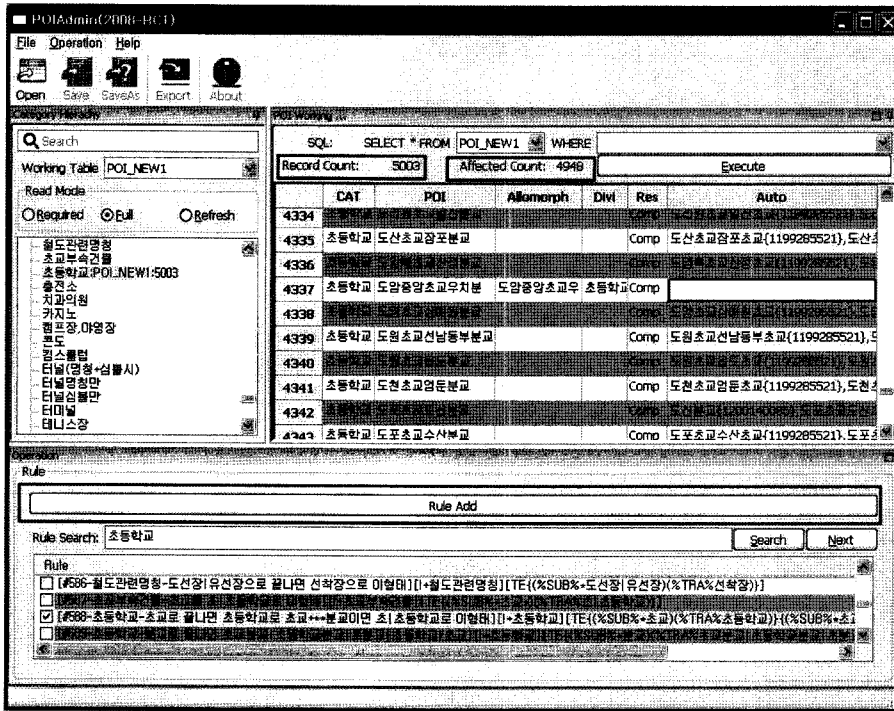


그림 11 POI 이형태 데이터베이스 구축 시스템

유는 원래의 DB에서 POI명이 잘못 등재되어 있는 경우와 규칙에 맞지 않는 POI 항목도 있기 때문이다. 이처럼 원 POI명이 잘못된 경우는 수작업으로 이형태를 생성해야 하며 이때에는 해당 항목의 "Allomorph" 필드에서 직접 추가하면 된다. 오른쪽 상단의 Where란에서는 SQL 쿼리 언어를 직접 입력하여 편집하고자 하는 항목만을 선택하여 검색할 수 있는 기능을 제공한다. 예를 들어, Res <> "COMP"와 같은 조건을 주면 작업이 완전히 끝났음을 확인하지 않은 개체들만을 대상으로 작업을 진행할 수 있는 환경이 된다. 중앙의 "Rule Add"는 규칙 편집기를 실행하기 위한 버튼이며 바로 아래의 텍스트 박스는 생성된 규칙들에서 검색하고자 할 때 이용되며 맨 하단 부분이 작성된 규칙들이다.

### 5. 실험 및 결과

본 논문에서 개발한 POI 이형태 데이터베이스 구축 시스템은 규칙에 의해 이형태의 자동생성이 가능하고 생성된 결과를 편집할 수 있는 반자동화 도구로서, RULE의 수정, 추가, 검색 그리고 이형태 생성 등의 기능을 포함하고 있다. 여러 규칙에 의해 동일한 이형태가 여러 개 생성되더라도 오직 한 개의 이형태만이 등록되며 삭제를 위해 Auto 필드에 생성 시 적용한 규칙의

번호를 표시해 두기 때문에 특정 규칙에 의해 생성된 이형태를 역 추적하여 삭제할 수 있는 기능도 가지고 있다. 본 시스템은 Windows-XP pro 환경에서 Visual C++ 2005를 이용하여 구현하였다.

#### 5.1 이형태 생성 통계

본 연구에서는 659,191개의 개체를 갖고 있는 POI DB를 대상으로 이형태 DB를 구축하였는데 구축 과정에서 동일 명칭의 POI는 단일화하여 중복을 제거한 393,443개의 POI명을 대상으로 이형태를 생성하였다. 자동화를 위해 450개의 패턴을 수집하였고 577개 규칙을 기술하였다. 물론 1~2개의 개체에 해당하는 경우도 규칙으로 기술할 수는 있지만 본 연구에서는 20개 이상의 개체에 영향을 미치는 경우에만 규칙으로 기술하였고 나머지는 수작업으로 이형태를 생성하였다. 중복이 제거된 상태에서의 이형태의 생성 비율은 47.79%였으며 이를 원래의 DB에 통합하였을 경우는 35.81%였다. 원래의 DB에서 더 낮은 비율을 보인 것은 중복적으로 많이 등재된 POI 명칭이 이형태가 거의 없는 2~5글자인 비율이 높았기 때문이다. 본 연구에서 이형태를 구축한 결과를 정리하면 표 1과 같다.

수작업으로 모든 이형태를 생성하는 것은 전체 엔트리에 대하여 반복적이고 지루한 작업이 계속되고 이에 따라 작업자의 실수 등이 결과에 포함되어 생산성이 현

표 1 생성된 이형태 구축 통계

구분	POI개체수	자동	수동	이형태 합계	비율
중복제거	393,443	113,946	74,061	188,007	47.79%
원 DB	659,191	197,742	80,449	248,981	35.81%

표 2 개체 당 생성된 이형태 수

이형태 수	자동	수동	합계	비율
1개	72,725	37,499	110,224	58.63%
2개	26,465	21,785	48,250	25.66%
3개	4,342	9,851	14,193	7.55%
4개	3,675	1,786	5,461	2.90%
5개	959	1,125	2,084	1.11%
6개	3,675	1,786	5,461	2.90%
7개	318	114	432	0.23%
8개이상	1,787	115	1,902	1.01%
총 갯수	113,946	74,061	188,007	

저히 떨어진다. 본 연구에서 기술한 577개의 규칙들을 통하여 총 197,742개 엔트리에 대하여 자동으로 이형태가 생성되었으며 이는 원 DB의 30%를 차지하는 양이다. 그렇지만 원 DB의 총 엔트리 중 이형태가 존재하는 것으로 조사된 수는 248,981개 이므로 실제적인 자동화율은 약 79%에 달한다. 따라서 수작업으로 이형태를 생성하는 것에 비하여 많은 시간과 생성 비용을 절감할 수 있다.

표 2는 POI별로 생성된 이형태의 수를 나타낸다. 이형태의 개수가 1-2 개인 경우가 84.29%이었으며 5개 이상 생성된 경우도 5.25%나 되었다. 규칙에 의해 자동으로 생성된 이형태는 수동에 의해 생성한 이형태의 약 1.5배였다. 또, 원 DB의 크기는 3.2MB이고 생성한 이형태를 모두 포함한 DB의 크기는 약 4MB이었다. 하나의 POI에 대해서 존재하는 이형태가 무한정이지 않기 때문에 검색시간 또한 거의 늘어나지 않는다.

5.2 검색 실험

“한국표준과학연구원”을 검색하기 위해 사용자는 “한국표준과학연구원”보다는 “표준과학연구원”이나 약칭인 “표준연” 또는 별칭인 “크리스”로 검색을 하게 된다. 그러나 현존하는 대부분의 DB 검색은 전방일치만을 지원하기 때문에 “한국표”로 입력해야만 이 검색어로 시작하는 “한국표준과학연구원, 한국표준기기, 한국표준협회, 한국표준협회연수원” 등의 POI 명칭이 검색된다. 이러한 단점을 개선하기 위해 개발된 본 시스템에서는 생략형이나 별칭, 약칭 등을 입력해도 해당하는 POI 명을 정확하게 검색해 준다.

다음의 그림 12는 “한국표준과학연구원”을 검색하기 위해 “한국표준과학연구원”의 생략형인 “표준과학연구원”, 일반적인 이형태 “표준연”과 “kriss”로 검색한 결과

를 보여주는 실험 화면이다. 이 그림에서 보여지는 것처럼 본 논문에서 제안한 방법으로 이형태를 생성하여 검색에 이용하면 다양한 키워드에 대해서도 검색이 가능하게 되기 때문에 보다 효율적이고 신뢰도가 높은 내비게이션 검색 시스템이 될 것이다.

검색 성능을 평가하기 위하여 대학생 100명을 대상으로 각각 20개의 POI 명칭에 대한 내비게이션 검색을 시도하도록 하였다. 실험 참가자는 내비게이션을 사용한 경험이 있고 POI명의 이형태에 대한 사전 지식이 전혀 없었다. 검색은 하나의 목적지에 대해서 단 1회의 검색만을 수행하고, 오타가 있는 경우에만 다시 입력할 수 있도록 하였으며 원 POI DB에 등록되어 있지 않은 POI명을 입력한 경우에는 무효 처리(실험의 결과에서 제외)를 하였다. 이러한 조건에서 실험한 결과, 총 2,000회의 검색에서 무효 11회를 제외하고 약 89%의 검색 성공률을 보였다. 이 결과를 분석해 보면 검색하고자 하는 POI명이 “전주영양돌솥밥”인 경우 “전주돌솥밥”이나 “영양돌솥밥”으로 검색하면 이형태를 사용하지 않는 경우는 검색에 실패했으나 본 시스템에서는 성공하기 때문에 검색 성공률이 차이를 보였다. 그러나 전주에 있는 덕진공원을 검색하기 위해 경우 POI 데이터베이스에 있는 POI명은 “덕진공원”으로 등록되어 있는데 사용자가 유추하여 “전주덕진공원”과 같이 POI명을 입력한 경우 검색에 실패하였다. 또 전발연(전북발전연구원)처럼 특정 지역에서 사용하는 약칭 등을 사용한 경우에도 실패하였다.

6. 결론 및 향후연구

현재의 내비게이션 시스템은 POI명을 검색하기 위해 POI명을 기준으로 전방일치 방법에 의한 검색 결과만을 이용하기 때문에 사용자의 검색 요구를 정확히 충족할 수 없게 된다. 따라서 본 논문에서는 POI 검색 서비스를 이용하는 사용자가 다양한 명칭으로 POI를 검색하고자 할 때, 사용자의 검색 요구를 충족할 수 있도록 POI에 대한 이형태 DB를 자동으로 구축해주는 시스템을 제안하였다.

시스템을 구축하기 위해 POI명을 분석하여 이형태 생성 방법을 7가지 유형으로 분류하고, 각 유형별 패턴을 찾아 자동화가 가능한 형태의 규칙으로 기술하였다. 이렇게 기술된 규칙을 기반으로 제약 조건 등을 추가적으로 기술하여 자동화 툴을 통하여 POI 명칭에 대한 이형태를 자동으로 생성하였다.

중복이 제거된 상태에서의 이형태의 생성 비율은 47.79%였으며 원DB에서는 35.81%였다. 이형태를 이용한 POI명 검색 실험에서는 평균 89%의 검색 성공률을 보였다. 이것은 내비게이션의 POI명 검색 서비스에 이



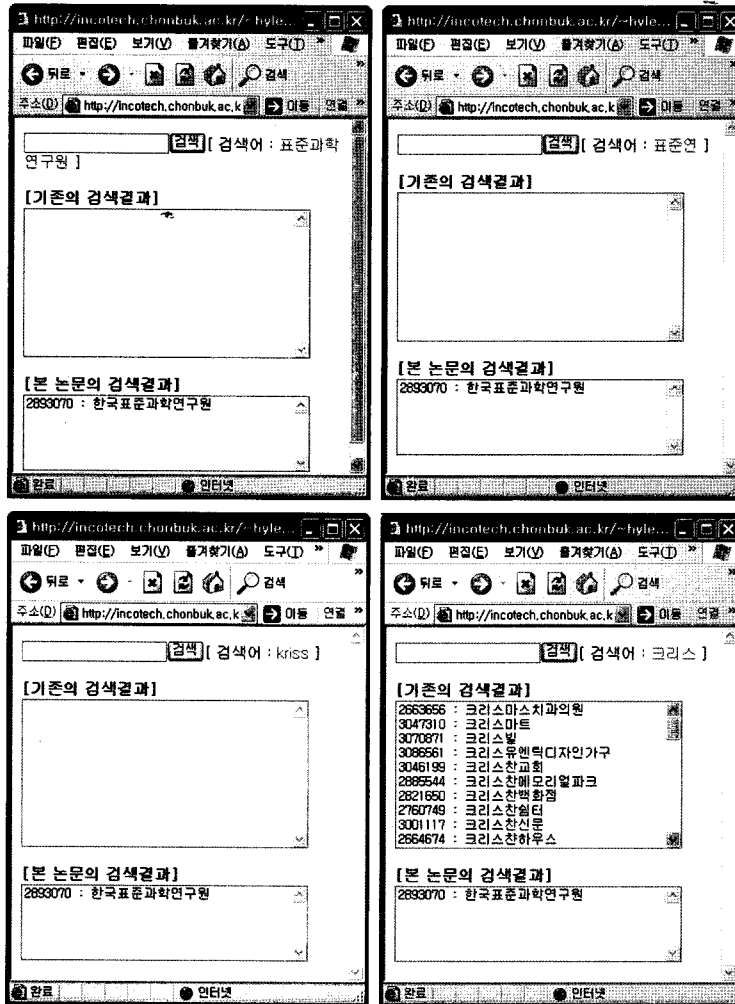


그림 12 검색 결과 비교

형태를 이용하게 되면 보다 효율적인 검색이 가능하게 됨을 의미한다.

현재 내비게이션에 음성인식 기능을 추가하려는 연구가 여러 곳에서 진행되고 있는 바 이들 연구가 구현되면, 사용자가 POI명을 발화할 때 이형태를 사용하는 빈도가 점점 높아질 것이므로 본 연구의 결과는 더욱 의미가 있다.

향후 연구과제로는 “덕진공원”처럼 POI명에 공원명이나 기관명만 있는 경우 그 명칭이 존재하는 지역명을 추가하여 “전주덕진공원”과 같은 이형태를 생성해야 하는지에 대한 검토가 필요하며 음성인식으로 인한 검색 시스템과 연계되기 위해서는 하나의 POI명에 대하여 정확한 이형태와 가급적 적은 수의 이형태를 생성해야만 전체적인 시스템의 성능을 향상시킬 수 있기 때문에 이

에 대한 연구가 필요하고 또한 원DB 자체의 오류를 줄이는 방안이 연구되어야 한다.

### 참 고 문 헌

- [1] 박용진, 김흠, 한원희, 서세광, 송은하, 한성국, 정영식, “계층적 POI 정보 가시화를 위한 GVP 개발”, 한국정보과학회, 2007 한국컴퓨터종합학술대회 논문집 제34권 제1호(B), pp. 390-395, 2007. 6.
- [2] 김대식, 김형진, 손봉수, 유완, “생활지리정보 검색 및 안내를 위한 POI의 구축 및 활용”, 한국콘텐츠학회/한국통신학회, 2003 추계 종합학술대회 논문집 제1권 제2호, pp. 423-430, 2003. 11.
- [3] 이정은, 신성현, 황현숙, 정신일, 김창수, “생활지리정보시스템의 PoI기반 정보등록 서비스 모델”, 한국콘텐츠학회, 2007 추계 종합학술대회 논문집 제5권 제1호, pp. 16-20, 2007. 6.

- [4] Yeo-Chan YOON, So-Young PARK, Young-In SONG, Hae-Chang RIM and Dae-Woong RHEE, "Automatic Acronym Dictionary Construction based on Acronym Generation Types," IEICE TRANS, INF & SYAT., 2007.
- [5] Yan Qu, Gregory Grefenstette, "Finding Ideographic Representation of Japanese Name Written in Latin Script via Language and Corpus Validation," ACL 2004.
- [6] 아이스스테이션 v43 Navi, <http://www.i-station.co.kr/>
- [7] 아이나비, <http://www.inavi.com/>



양 승 원

1995년 전북대학교 전산통계학과(이학박사). 2003년~2004년 Univ. of GUELPH (CANADA) 방문교수. 1994년~현재 우석대학교 게임콘텐츠학과 교수. 관심분야는 자연언어처리, 정보검색, 음성처리



이 현 영

1991년 전북대학교 전산통계학과 졸업  
1996년 전북대학교 전산통계학과 이학석사, 2008년 전북대학교 전산통계학과 이학박사. 관심분야는 한국어처리, 정보검색, 데이터 색인



왕 지 현

1996년 충북대학교 컴퓨터공학과(학사)  
1998년 충북대학교 컴퓨터공학전공(공학석사). 2000년~현재 한국전자통신연구원 음성처리연구팀. 관심분야는 자연언어처리, 정보검색, 기계학습