

# 구문 제약으로 문형을 사용하는 CFG기반의 한국어 파싱

박인철<sup>1\*</sup>

## CFG based Korean Parsing Using Sentence Patterns as Syntactic Constraint

In-Cheol Park<sup>1\*</sup>

**요 약** 한국어는 용언이 의미적 제약을 통해 문장을 지배하며 대부분의 한국어 문장은 주절과 내포문을 가지는 복문으로 구성되어 있다. 따라서 한국어에 맞는 구문 문법이나 구문 제약을 기술하는 것은 매우 어렵고 한국어를 파싱하면 다양한 구문 모호성이 발생한다. 본 논문에서는 구문 제약으로 문형(sentence patterns)을 사용하는 CFG기반의 문법을 기술하여 구문 모호성을 해결하는 방법을 제안한다. 이를 위해 내포문을 포함하는 복문도 문형으로 분류하였으며 44개의 문형을 사용한다. 그러나 한국어 특성상 문형 정보만으로는 모든 구문 모호성을 해결할 수가 없기 때문에 문형에 의미 제약(semantic constraint)을 가한 의미 지표(semantic marker)를 사용하여 파싱을 수행한다. 의미 지표는 보조사의 처리나 공동격 조사에 의해 발생하는 구문 모호성을 해결하는데 이용될 수 있다.

**Abstract** Korean language has different structural properties which are controlled by semantic constraints of verbs. Also, most of Korean sentences are complex sentences which consisted of main clause and embedded clause. Therefore it is difficult to describe appropriate syntactic grammar or constraint for the Korean language and the Korean parsing causes various syntactic ambiguities. In this paper, we suggest how to describe CFG-based grammar using sentence patterns as syntactic constraint and solve syntactic ambiguities. To solve this, we classified 44 sentence patterns including complex sentences which have subordinate clause in Korean sentences and used it to reduce syntactic ambiguity. However, it is difficult to solve every syntactic ambiguity using the information of sentence patterns. So, we used semantic markers with semantic constraint. Semantic markers can be used to solve ambiguity by auxiliary particle or comitative case particle.

**Key Words** : CFG based grammar, PATRII, Sentence patterns, Syntactic ambiguity

### 1. 서론

한국어는 생략이 자주 발생하고 부분 자유 어순을 갖는 비구조적 언어이며 용언이 의미적 제약을 통해 문장을 지배한다. 또한 한국어는 상황 중심의 언어라 불릴 만큼 의미와 화용(discourse)이 중요한 역할을 한다.

따라서 한국어는 영어와는 달리 정교하게 설계된 문법 규칙과 엄밀한 제약을 갖는 문법 이론보다는 규칙외적인 정보에 의해서 분석이 주도된다[6]. 이러한 관점에서 볼 때 한국어의 파싱을 위해서는 규칙은 간단히 기술하고 분석 도중에 각 형태소(morpheme)들의 문법적 관계를 검

사하면서 분석하는 방법이 타당하다고 볼 수 있다.

기존의 파싱 방법론은 서양 언어의 분석 틀을 이용하여 한국어를 분석하려고 하였다. 단일화 기반의 문법 이론[2, 6]들은 비문(ungrammatical sentence)을 가려내기 위하여 각종 제약 조건들을 설정하여 분석하는 방법이다. 그러나 어순이 자유롭고 의미가 중요한 역할을 하는 한국어 분석에 문장 성분을 직접 적용하기에는 어려웠다. 또한 의존 문법[7]은 생략이 빈번하고 어순이 자유로운 특성을 반영하기 위해 개발되었지만 구조 정보에 의해 간단히 제거될 수 있는 파스들도 분석의 대상이 되기 때문에 너무 많은 파스 트리가 생성된다. 이런 이유로 현재

본 논문은 2008학년도 호원대학교 교내학술연구구조성비의 지원에 의하여 연구되었음

<sup>1</sup>호원대학교 컴퓨터게임학부 교수

접수일 08년 4월 6일

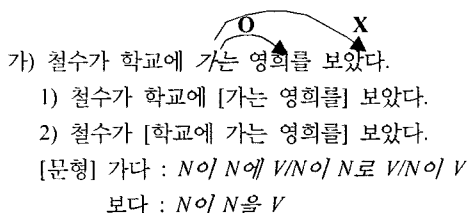
수정일 08년 6월 24일

\*교신저자: 박인철(icpark@howon.ac.kr)

재제확정일 08년 8월 11일

까지도 한국어 파싱의 표준이 될 수 있는 방법론이 대두되고 있지 않은 실정이다.

파싱은 문장에서 명사구와 용언 사이의 표층적 문법 관계를 밝히는 작업이다. 파싱 단계에서 발생하는 대부분의 모호성은 용언과 체언이나 부사구의 결합에 따라 나타난다. 즉, “NP + VP”나 “ADVP + VP”, “VP + NP”로 결합할 때 발생한다. 예를 들어, 아래의 가) 문장에서 “학교에”라는 체언구는 용언 “가다”나 “보다”에 모두 부착할 수가 있지만 “가다”의 문형 정보 “N이 N에 V”라는 문형 정보를 사용하면 “가다”에 부착되는 것이 올바른 분석임을 알 수가 있다.



이와 같이 한국어에서는 용언이 요구하는 조사의 유형과 명사나 부사의 유형이 존재한다. 이러한 문장이 갖는 구조적 유형을 문형(sentence patterns)이라고 한다(4). 용언의 하위범주화 정보를 한국어의 특성에 맞게 재분류한 문형을 제약 조건으로 사용함으로써 구문 모호성을 해결할 수 있으며, 또한 문형 정보만으로 구문 모호성을 해결할 수 없는 경우에는 문형에 의미제약을 가한 의미지표를 이용하여 모호성을 해결할 수 있다.

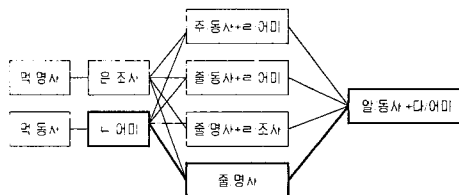
본 논문에서는 한국어의 파싱에서 발생하는 구문 모호성의 원인을 파악하고 이를 해결할 수 있는 방법을 제안한다. 구문 모호성의 원인은 크게 두 가지로 분류할 수가 있는데 첫째는 파싱의 전단계인 형태소 분석 결과에서 발생하는 형태론적 모호성이다. 이것은 [10]이 제안한 구문 형태소(syntactic morpheme)를 이용하여 어느 정도는 해결할 수가 있다. 둘째는 한국어의 특성상 발생하는 모호성이다. 이를 해결하기 위하여 한국어의 특성을 살펴보고 모호성을 해결할 수 있는 방안으로 문형을 구문 제약 조건으로 사용하는 CFG 기반 한국어 파싱 방법을 제안한다.

## 2. 파싱 관점의 한국어 특성

### 2.1 형태론적 특성

한국어는 형식 형태소가 많이 발달하였고 여러 형태소

들이 결합하여 하나의 구문적 단위를 이루는 경우가 많다. 이러한 형태소열은 형태론적 모호성과 구문 모호성의 원인이 된다. 따라서 이를 해결하기 위한 많은 연구가 진행되어 왔다(3, 5, 7, 8, 10). 특히, [10]은 기능적 단위의 형태소만을 하나의 결합 단위로 하는 구문 형태소를 제안하였으며 구문 형태소는 구문 해석을 위한 하나의 단위가 되기 때문에 구문 해석의 효율을 높일 수 있었다.



[그림 1] “먹은 줄 알다”의 형태소 결과

위의 [그림 1]은 “먹은 줄 알다”에 대한 형태소 해석 결과이다. [그림 1]에서 보는 바와 같이 우리는 8개의 형태론적 모호성이 발생함을 알 수 있다. 그러나 [10]이 제안한 구문 형태소를 이용하면 “- 줄 알다”라는 형태소열이 결합하여 ‘추측’이라는 양상 정보(modality)로 표현된다. 따라서 “먹다/pvg [추측]”라는 하나의 결과만을 얻을 수가 있다. 이와 같이 구문 형태소는 형태론적 모호성을 해결하는데 이용될 수 있음을 알 수 있다. 따라서 본 논문에서는 구문 형태소를 구문 해석의 입력 데이터로 사용한다.

### 2.2 구문적 특성

한국어는 생략이 빈번하고 자유 어순을 갖는 비구조적 언어이다. 또한 용언에 따라 다양한 격조사와 부사를 요구한다. 따라서 문장의 구조를 파악하기 위해서는 정형화된 구문적 정보만을 이용할 수는 없다. 예를 들어, 다음의 문장들을 살펴보자.

- 1) 철수가 귀찮게 군다.
- 2) 철수가 군다.\*
- 3) 철수가 순이를 대표로 뽑다.
- 4) 철수가 순이를 대표에게 뽑다.\*

1)과 2)에서 ‘군다’는 자동사이므로 주어만을 필수성분으로 요구함을 알 수 있다. 그러므로 1)과 2)는 구문적으로 옳은 문장으로 분석이 된다. 그러나 ‘군다’라는 용언은 ‘어떠하게’라는 의미를 가지는 부사를 문장의 필수성분으로 요구한다. 따라서 2)는 의미적으로 올바른 문장이 아님을 알 수가 있다. 또한 3)과 4)에서 ‘뽑다’는 “~로”

라는 조사가 올 수 있지만 “~에게”라는 격조사는 타당하지 않다. 이러한 현상은 ‘근다’나 ‘뵈다’라는 용언에만 한정된 것이 아니다.

이와 같이 한국어는 특별한 격을 수반하는 용언들이 많이 존재한다. 이러한 용언의 경우 나머지 격을 보조적인 의미로 파악하기 때문에 문장의 올바른 의미를 파악하기 어렵거나 모호성 발생의 원인이 된다. 따라서 이러한 용언들의 구조적 유형을 문형이라는 틀로 제약할 필요가 있다. 앞의 예문들에서 보느냐와 같이 한국어의 경우 구분 분석에서 이러한 문형 정보의 이용은 필수적임을 알 수 있다.

**2.3 구조적 특성**

한국어 문장은 용언 중심의 언어로 모든 체언구나 부사구는 용언의 지배를 받는다. 따라서 아래의 예문 나)와 같이 단문인 경우에는 문장에서 사용된 체언구나 부사구를 모두 용언에 부착하며 구문 모호성은 발생하지 않는다.

나) 철수가 아침에 학교에서 밥을 맛있게 먹었다.

[문형] : N이 N을 먹다.

또한 중문에서는 생략이 문제가 될 수 있지만 전체 주절에 대한 처리를 할 때 생략된 부분을 보완하는 방법이 [9]에 의해 제안되었다. 예를 들어, 예문 다)에서는 용언 “먹다”의 문형에 의해 “철수가”가 “먹다”의 주어로 인식을 하면 “부르다”의 주체는 생략되었음을 알 수가 있다. 이 경우는 전체 주절의 의미상 주어 “철수가”로 설정하며 의미분석이나 기계번역에서는 이를 활용한다.

본 논문에서는 “철수가 밥을 먹고 그리고 철수가 노래를 부르다”의 의미를 가지도록 “(철수가 밥을 먹고) [철수가] 노래를 부르다”로 분석을 수행한다. 그러나 문두에 나와 있는 “철수가”의 격조사가 “은/는” 등으로 바뀌면 “철수는 밥을 먹고 나서 철수가 노래를 부르다”의 의미를 가지도록 “철수는 (밥을 먹고) 노래를 부르다”로 분석을 수행한다.

다) (철수가 밥을 먹고) 노래를 부르다.

다’) 철수는 (밥을 먹고) 노래를 부르다.

[문형] : N이 N을 먹다.

N이 N을 부르다.

그러나 주절과 내포문으로 구성되는 복문은 내포문의 범위를 어떻게 결정하느냐에 따라 체언구나 부사구 부착의 문제가 발생된다. 이를 해결하기 위하여 문형 정보를 제약조건으로 사용하는 단문으로 분할하는 방법을 기반

하여 내포문의 범위를 정하고 주절과의 관계와 기능에 따라 명사절, 관형절, 부사절로 분류한다. 이를 정리하면 다음 [표 1]과 같다.

[표 1] 내포문의 구분

| 구분  | 세분화           |
|-----|---------------|
| 명사절 | 인용 명사절        |
|     | ‘모/기’ 명사절     |
|     | ‘지/나’ 명사절     |
| 관형절 | 관계 관형절        |
|     | 동격 관형절        |
|     | 의존 관형절        |
| 부사절 | 부사성 활용 어미에 따라 |

**3. 제약 조건으로 문형을 사용하는 CFG 기반의 문법**

**3.1 문형 정보**

한국어에서 문장의 중심은 용언이기 때문에 용언을 기준으로 문형을 설정한다. 문형을 설정하기 위해서 [1, 4]를 활용하였으며 사용하는 문형은 동사가 31개, 형용사가 8개, 서술격 동사가 5개로 총 44개이다. 다음 [표 2]와 [표 3]은 본 논문에서 분류한 문형의 종류와 사용 빈도가 높은 문형의 일부를 나타낸 것이다.

[표 2] 분류한 문형의 일부

| 동사 문형                    | 형용사 문형                 |
|--------------------------|------------------------|
| V1) N(이/는/은/가)+V         | A1) N(이)+A             |
| V2) N(이)+N(에/에게)+V       | A2) N(이)+N(에)+A        |
| :                        | :                      |
| V11) N(이)+N(를)+V         | A8) N1(이)+N(와)+N2(이)+A |
| V12) N(이)+N(에/에게)+N(를)+V | N1) N이 N이다             |
| :                        | :                      |
| V30) N[이]+N[와]+N[에]+V    | N5) N1이 N2이 N이다        |
| V31) N[이]+N[에게]+N[에서]+V  |                        |

[표 3] 상위 빈도의 문형정보

| 동사문형 | 빈도     | 형용사문형 | 빈도     |
|------|--------|-------|--------|
| V1   | 41.29% | A1    | 18.28% |
| V11  | 34.49% | A5    | 2.19%  |
| V2   | 13.96% | A2    | 1.9%   |
| V12  | 5.98%  |       |        |

### 3.2 PATRII의 확장

조건단일화 기반 PATRII 문법은 문장의 구조를 간단히 기술하고 문장 성분간의 관계는 동적으로 검사하면서 분석할 수 있는 방법이다[6].

본 논문에서는 문형 정보를 이용한 CFG 기반의 문법을 작성하고 이를 구문 분석을 위한 LR 파싱 테이블과 조건 제약을 위한 함수로 번역하여 LR 파서 기반의 구문 분석기로 이용한다. 또한 문형을 이용하여 문맥 의존적인 특성을 반영하고 이러한 검사를 조건 단일화라는 제약 조건을 통해서 행할 수 있도록 단일화 식을 작성한다. 따라서 기존의 PARTII 문법에서는 무조건적인 순수 단일화 이론만을 택하고 있기 때문에 [6]이 제안한 확장된 PARTIII를 사용하여 문법을 기술한다. 그러나 기존의 확장된 PATRII 문법에 사용된 연산자만으로는 한국어의 다양성을 모두 적용하는데 한계가 있어 아래와 같이 단일화 연산자와 조건 체크 명령어를 추가하여 사용한다.

#### 1) 조건 체크 명령어의 추가

- a) \*member-not\* : 여러 개의 문형 정보를 평가할 때 사용  
 사용 예 : ((x0 sp-info) =c (\*MEMBER-NOT\* V4 V10))  
 의미 : LHS에 있는 문형정보(sp-info)가 V4, V10형이 아니면 단일화를 수행하라는 의미
- b) \*eor\* : 여러 개의 단일화 식에서 처음으로 조건에 맞는 항목 하나만 선택하여 실행  
 사용 예 : (\*eor\* ((x1 mm-mod) = \*defined\*)  
 (((x1 mm-mod) = \*undefined\*)))  
 의미 : x1의 mm-mod가 정의되어 있으면 앞의 것만 실행하고 정의되어 있지 않으면 뒤의 것을 실행 - \*or\*는 모든 단일화 식을 수행

#### 2) 단일화 연산자의 추가

- c) =e/=n : 동일성 체크 기능으로 =e(equal), =n(not equal) 기능을 수행  
 사용 예 : ((x0 sm-info) =e (x1 sm-info))  
 의미 : x0의 문형정보(sp-info)와 x1의 문형정보가 같은지만 검사 - 단일화를 수행하지는 않음
- d) =a : 오른쪽(RHS)에 있는 단일화 식을 평가한 후 그 값을 LHS에 할당할 때 사용  
 사용 예 : ((x0 svalue) =a (- (x0 svalue) 8))  
 의미 : x0의 svalue 값에서 8을 뺀 후(RHS) 그 값을 x0의 svalue 값으로 설정(LHS)

### 3.3 조건 단일화 기반 CFG

단일화 기반 문법 이론은 특정 언어에 대하여 비문을 가려내기 위한 각종 제약 조건들을 설정하여 분석의 일관된 틀을 제시하는 것이다. 조건 단일화는 [표 4]와 같이 규칙의 명세를 'IF-Then-Else' 형태로 기술하여 자질 구조안의 특성 속성 값을 필요한 시점에서 검사함으로써 각 문장 성분이 가지는 문법적 관계를 검사하여 조건에 맞는 단일화만을 수행한다는 전략이다.

본 논문에서는 한국어 파싱을 위한 기본 틀로 문형을 제약조건으로 사용할 수 있는 조건 단일화 기반의 CFG를 이용한다.

이는 단어들이 결합하여 문장을 이루는 경로(path)로 구조 규칙을 설정하고, 개개의 구조조들이 결합할 때 조건 단일화 기반의 제약에 의하여 구조조의 결합을 제약하는 방법이다. 이는 구조조 규칙의 간접합과 구조조의 의존적 언어의 특성을 조건 단일화 제약을 통해 문장을 분석하는 방법이다.

한국어는 관형절을 포함하는 문장이 많다. 예를 들면, 다음 [표 4]에서 <VNP> <=> <NPS> <VNP>는 관형절을 처리하기 위한 단일화 기반 구조조 규칙으로 'VNP'는 '먹은 사과'와 같은 구조를 갖는 관형절을 의미하며 'NPS'는 '철수가'와 같이 주격 조사를 갖는 체언구를 의미한다. 따라서 이 문법 규칙에 의해 만들어지는 구조조는 '철수가 먹은 사과'와 같은 형태가 된다. 이 문법에서 제약정보 'sp-info'는 문형을 의미하며 'sm-info'는 의미지표를 의미한다.

[표 4] 구문제약으로 문형을 사용한 CFG 기반 문법

```

<VNP> <=> <NPS> <VNP>   ::: CFG rule
(*or*
  ((x2 topic) =c subj
   (x2 sp-info) =c v6       :: SPI constraint
   (x2 subj) = *undefined*
   (x2 comp) = *undefined*
  (*or*
    ((x1 sm-info) =c ANI   :: SM constraint
     (x2 subj) = x1)
    ((x1 sm-info) =c (*not* ANI)
     (x2 comp) = x1))
  :
(x2 vns) = +
x0 = x2
    
```

이와 같이 구문 모호성을 해결하기 위하여 CFG 기반

의 조건 단일화를 이용하여 파싱을 수행한다. 또한 문형은 용언에 따른 NP의 문법 형태소 정보를 파악하기 위한 제약으로 사용하며 의미 지표는 문형에 대한 제약으로 사용한다.

### 3.4 한국어 파싱 문법의 기술

한국어를 구문 분석하기 위해서 동사, 형용사, 서술격 동사로 용언을 세분하여 문형 사전을 구축하였고, 조건 단일화 기반의 CFG를 이용하여 문법 규칙을 기술하였다. 또한 관형형 어미를 가지는 용언은 후행하는 체언구와 먼저 단일화 연산을 수행하도록 하기 위해서 일반 용언과 분류하여 기술하였다. 이렇게 기술된 문법 규칙은 총 120개이다. 일반적인 구구조 기법보다 문법 규칙의 수가 적은 것은 어절 단위의 분석과 구문 형태소, 자질 정보, 대분류된 품사 태그를 이용하고 문법 규칙은 이진 문법으로만 기술하였기 때문이다. 본 논문에서 사용한 문법 규칙의 일부는 [표 5]와 같다.

[표 5] 문법 규칙의 예

|                      |            |
|----------------------|------------|
| <PAM> <=> <PA>       | ::예쁜       |
| <ANP> <=> <PAM> <NP> | ::예쁜 철수    |
| <ANP> <=> <NP> <ANP> | ::눈이 예쁜 철수 |
| <NP> <=> <ANP>       | ::눈이 예쁜 철수 |
| <XP> <-- (@np)       | ::철수       |

## 4. 실험 및 평가

### 4.1 실험 및 분석

실험을 위해 KIBS[1]에서 단문으로 분할이 가능한 10 어절 이내의 700문장을 추출하였다. 추출된 문장은 평균 3.55개의 용언(보조 용언은 제외)을 포함하는 복문의 구조를 가지고 있다. 구문 모호성을 제약하기 위해서 조건 단일화 기반으로 하는 일반적인 구문분석 방법(실험1)[6], 구문형태소와 문형을 제약조건으로 사용하는 방법(실험2)[10], 본 논문에서 제안한 문형과 의미적 제약조건을 이용한 방법(실험3)을 대상으로 비교 실험하였다.

추출된 700개의 입력문장에 대해 각각의 실험 결과에 따른 구문 분석 결과의 평균 파스 트리의 수(구문 모호성의 수)는 [표 6]과 같다.

[표 6] 실험에 따른 구문 모호성의 수와 소요시간

| 구분         | 실험1    | 실험2    | 실험3   |
|------------|--------|--------|-------|
| 평균 파스 트리 수 | 68.43개 | 18.21개 | 6.93개 |
| 소요 시간      | 2.06초  | 1.21초  | 1.09초 |

실험에 사용된 컴퓨터는 Pentium4이며 CPU는 2.4GHz 이고 RAM은 512MB를 사용하였다. 실험에 있어 700개 문장에 대한 각각의 입력 문장을 구문 분석하는데 소요된 시간은 평균적으로 [실험1]이 2.06초, [실험2]가 1.21초, [실험3]은 1.09초가 소요되어 제안한 방법이 효율적이었다.

또한 구문 분석한 결과 중에서 올바른 후보가 생성되지 않은 경우는 9개 문장이었다. 예를 들어, 입력 문장 “꽃이 아름다운 공원을 보았다”에 대해 “아름답다”의 문형 정보(N이 아름답다)만을 이용하면 “꽃이 [아름다운 공원을] 보았다.”와 “[꽃이 아름다운 공원을] 보았다.”로 분석되었으며 의미적으로는 “[꽃이 아름다운 공원을] 보았다.”로 분석되는 것이 타당하다.

이 문제를 해결하기 위해서는 “N이 N이 아름답다”라는 새로운 문형 정보를 추가하는 것을 고려해 볼 수가 있다. 그러나 이 경우에는 “철수가 아름다운 공원을 보았다.”라는 문장에 대해 “[철수가 아름다운 공원을] 보았다.”로 해석되는 구문 모호성을 초래하며 이는 문형 정보만에 의한 구조적인 문제로 공기정보를 이용하여 해결해야 한다.

### 4.2 실험 평가

실험 결과를 분석해 보면 신문이나 교과서보다는 소설에서 발췌한 문장에서 오류가 많았다. 이는 소설의 경우 어순의 도치와 생략이 심하고 “철수의 손잡이가 달린 가방”과 같이 의미 정보가 요구되는 문장이 많았기 때문이다.

명사들의 나열인 경우에도 명사구의 범위에 의해 많은 모호성이 발생하였고, 보조사에 의한 불확실한 격 정보 때문에 많은 구문 모호성이 발생하였다.

그러나 제안한 방법으로 구문 분석을 수행하면 구문 모호성의 수가 평균 68.43개에서 6.93개로 줄어들었음을 알 수 있었으며 이는 실험 문장이 대체로 10어절 내외의 문장임에도 불구하고 문형을 제약조건으로 한 단문으로 분할하여 분석하는 방법이 한국어의 구문 분석에 매우 효율적임을 알 수 있었다.

## 5. 결론

본 논문에서는 부분 자유 어순과 기능어가 발달된 한국어를 구문분석하기 위해서 한국어 문장을 문형으로 분류하고 구문 제약 조건으로 문형을 이용하는 CFG기반의 조건단일화 문법을 기술하였다.

문형은 내포문을 포함하는 복문에서 체언구나 부사구

의 처리, 이중주어나 이중목적어의 처리, 공동격 조사 ‘와’ 등을 파싱하는데 좋은 제약이 되었다. 또한 문형 정보만으로 해결되지 않는 구문 모호성은 문형의 필수격에 의미를 제약하여 해결하였다. 이러한 문형 정보를 기반으로 한국어 문장을 분석하는 방법은 정보 검색이나 기계 번역, 자연어 이해와 같은 응용 시스템에 보다 효율적으로 적용될 수 있을 것이다.

향후 연구방향으로는 분류한 문형을 좀 더 세분화하고 한국어의 여러 현상을 처리할 수 있도록 파싱 문법을 개선하는 것이다.

## 참고문헌

- [1] KIBS : Korean Information Base System, <http://kibs.kaist.ac.kr/kibs>
- [2] Tomabechi, H., "Efficient Unification for Natural Language," Doctoral dissertation, Carnegie Mellon University, 1993.
- [3] 강승식, 음절 정보와 복수어 단위 정보를 이용한 한국어 형태소 분석, 서울대학교 박사학위 논문, 1993.
- [4] 강은국, 조선어 문형 연구, 박이정출판사, 1996.
- [5] 김창제, 정천영, 김영훈, 서영훈, “부분적인 어절 결합을 이용한 효율적인 한국어 구문 분석기”, 정보과학회 가을 학술 발표논문집, pp.597-600, 1995.
- [6] 양승원, 조건 단일화 기반 PATRII를 이용한 한국어 구문 분석, 전북대 박사 학위 논문, 1995.
- [7] 윤덕호, 김영택, “다단계 여과 및 탐색을 이용한 의존 문법에 기반을 둔 한국어 분석 알고리즘”, 한국 정보과학회 논문지, Vol. 19, No. 6, pp. 614-624, 1992.
- [8] 이희자, “현대국어 관용구의 결합 관계 고찰”, 제6회 한글 및 한국어 정보처리 학술대회, pp.333-352, 1994.
- [9] 장재철, 박의규, 나동렬, “구간 분할 기반 한국어 대등 접속 구문분석 기법”, 제 14회 한글 및 한국어 정보처리 학술대회, pp.139-146, 2002.
- [10] 황이규, 이현영, 이용석, “형태소 및 구문 모호성 축소를 위한 구문 단위 형태소의 이용”, 한국정보과학회 논문지, Vol. 27, No. 7, pp. 784-793, 2000.

## 박인철(In-Cheol Park)

[정회원]



- 1984년 2월 : 전북대학교 전산 통계학과(이학사)
- 1986년 2월 : 전북대학교 전산 통계학과(이학석사)
- 1998년 8월 : 전북대학교 전산 통계학과(이학박사)
- 1992 ~ 현재 : 호원대학교 컴퓨터게임학부 교수

### <관심분야>

한국어정보처리, 정보검색, 의미지식표현, 시맨틱웹, 유비쿼터스 컴퓨팅