# 복합 레이블을 적용한 한국어 구문 규칙
## (Korean Syntactic Rules using Composite Labels)

김 성 용 † 이 공 주 †† 최 기 선 †††
(Seongyong Kim) (Kong Joo Lee) (Key-Sun Choi)

**요 약** 본 논문에서는 한국어 구문 분석 및 구문 트리 표현을 위한 복합 레이블 생성 방법을 제안한다. 기존의 구문트리 표현에서는 미리 정의된 구문 트리 레이블을 사용하여 구문 정보를 표현하였다. 본 논문에서는 이진규칙하에서 품사태그 정보만을 이용하여 구문 레이블을 자동으로 생성하는 방법을 제시한다. 제안된 구문 레이블은 두 개의 하위 구성체의 품사정보를 적절히 구성하여 형성되며, 동시에 현 구성체의 상태 및 역할 정보를 표현할 수 있도록 고안되었다. 이와 같이 함으로써 품사태그 정보가 가지고 있는 정보를 그대로 구문 트리에 반영시킬 수 있었다. 또한, 품사 정보와 이진규칙만을 이용하여 구문트리를 표현하기 때문에, 다양한 구문 규칙을 채택하고 있는 서로 다른 구문 분석기의 결과를 정규화하는 데 적용할 수 있을 것이며, 일본어와 같은 다른 언어에도 쉽게 적용 가능하다. 약 31,080 문장에 대한 구문 분석의 결과, 79.30%의 정확도를 얻을 수 있었으며, 이는 제안된 구문트리 표현 방법이 구문 분석기의 효율에도 좋은 영향을 미침을 보이는 것이다.

**키워드** : 한국어 구문 분석, 이진규칙, 복합 레이블링, 레이블링 알고리즘

***Abstract*** We propose a format of a binary phrase structure grammar with composite labels. The grammar adopts binary rules so that the dependency between two sub-trees can be represented in the label of the tree. The label of a tree is composed of two attributes, each of which is extracted from each sub-tree so that it can represent the compositional information of the tree. The composite label is generated from part-of-speech tags using an automatic labeling algorithm. Since the proposed rule description scheme is binary and uses only part-of-speech information, it can readily be used in dependency grammar and be applied to other languages as well.

In the best-1 context-free cross validation on 31,080 tree-tagged corpus, the labeled precision is 79.30%, which outperforms phrase structure grammar and dependency grammar by 5% and by 4%, respectively. It shows that the proposed rule description scheme is effective for parsing Korean.

**Key words** : syntactic analysis, Korean, agglutination, binary rules, composite label, automatic labeling algorithm

## 1. 서 론

한국어는 첨가어로서, 문장은 띄어쓰기에 의해 구분되는 어절로 이루어지고 어절은 여러 개의 형태소로 이루어지는데, 하나 이상의 실질형태소(lexical morpheme)에 0개 이상의 형식형태소(grammatical morpheme)가 결합하여 구성된다. 그러므로, 한국어 구문 분석을 위한 규칙을 기술하기 위해서는 어절과 형태소 중 어느 것을 규칙 기술의 단위로 할 것인지를 먼저 선택하여야 한다.

(1) 시간/ncn+이/jcs 귀중/ncps+하/xsm+ㄴ/etm
것/nbn+이/jp+다/ef+./sf

문장 (1)은 문장부호를 포함하여 3개의 어절과 9개의 형태소로 구성되어 있다. 형태소간의 결합은 '+'로 표시되었으며, 어절간의 띄어쓰기는 공백으로 표현되었다. 문장에서 밑줄 친 서술격조사 '이/jp'는 명사구 "귀중한 것"과 결합하는데, 이는 '것/nbn'과 '이/jp'가 단순하게 결합하는 것이 아니라 관형어 "귀중한"과 명사 '것/nbn'이 구문적으로 하나의 명사구를 생성한 이후에 그 명사구와 결합하는 것이다. 한국어는 지배소 후위의 언어이기 때문에 명사구 구성체(construct)와 결합한 서술격조사는 지배소 역할을 하며, 결과로 생성된 구성체는 문장에서 술어의 역할을 한다. 이러한 현상, 소위 후통사적

† 비 회 원 : 국방과학연구소 연구원
sykim@csone.kaist.ac.kr
†† 정 회 원 : 이화여자대학교 컴퓨터학과
kjlee007@ewha.ac.kr
††† 종신회원 : 한국과학기술원 전산학과
kschoi@cs.kaist.ac.kr
논문접수 : 2003년 8월 4일
심사완료 : 2004년 11월 5일

형태소 결합(post-syntactic morpheme concatenation: PSMC) 현상은 앞선 두 개의 구성체가 하나의 명사구 구성체로 생성된 다음에 서술격조사가 첨용되기 때문에 발생하게 된다.

한국어의 PSMC 현상은 어절을 분석 단위로 하는 규칙으로는 처리하기 어렵다. 따라서, 어절 단위의 규칙을 사용하는 연구들에서는 PSMC현상을 처리하기 위한 방편으로 기본 코퍼스(source corpus)와 입력 문장상의 어절 중 해당 현상이 발생하는 어절에 대하여 이 현상을 처리할 수 있도록 어절 분리 등의 재구성을 하게 되는데[1,2], 이는 기본 코퍼스를 왜곡하는, 바람직하지 못한 현상이다. 이러한 제한점으로 인해 한국어 구문 분석을 위해서는 형태소를 구문 분석의 단위로 하는 형태소 기반 규칙을 사용하는 것이 필요하다.

한국어 형태소는 명사, 동사어간과 같이 문장에서 상태(state)를 나타내는 실질형태소와 조사 및 어미와 같이 문장에서 구성체의 문법적 역할(role)을 담당하는 형식형태소로 나누어진다. 이 두 가지의 형태소는 문장을 구성하는 데 있어 서로 구분되는 쓰임새를 가지므로, 한국어 구문 규칙을 다룰 때에도 이 두 가지에 대한 구분이 있어야 한다.

이 논문에서는 형태소 단위의 규칙을 사용하며, 품사태그를 이용하여 구문 레이블을 자동으로 생성하는 알고리즘을 제시한다. 형태소 단위의 규칙을 사용함으로써 PSMC 현상을 다룰 수 있고, 품사태그 정보로부터 구문 레이블을 직접 생성함으로써 품사태그가 가지고 있는 정보를 그대로 활용 가능하다. 각 구문 레이블은 *"Det-Noun → Det Noun"*에서 *DetNoun*이 두 개의 하위 구성체인 정관사(*Det*)와 명사(*Noun*)의 레이블로부터 생성되는 형태와 같이 구성한다. 각 레이블은 문장에서 그 구성체가 가지는 상태 및 역할을 가능한 한 동시에 나타낼 수 있도록 구성한다. 이렇게 함으로써 구성체의 레이블이 하위 구성체의 정보뿐만 아니라 자신의 상태/역할 정보도 나타낼 수 있게 된다. 이러한 형태의 규칙 형식은 한국어 구문 분석을 위해 자주 사용되는 의존문법(dependency grammar, DG)과 매우 유사하므로, 의존 문법의 형태로도 적용 가능하다. 또한, 제안된 규칙 형식은 이진 규칙이라는 최소화된 형태로서, 이를 적용하면 다양한 규칙 표현들을 정규화(normalization)할 수 있다. 이를 통하여 임의의 코퍼스로부터 정규화된 코퍼스를 만들 수 있고, 서로 다른 문법 형식에 따른 상이한 트리 구조들을 상호 비교할 수 있다.

## 2. 기존 연구

한국어는 부분 자유 어순인 첨가어로서 구문 분석을 위하여 다양한 문법이 사용되고 있다. 그 중 의존문법[1,2]과 구구조문법[3,4]이 주로 적용되며[5], 최근에는 결합범주문법(CCG)[6, 7]도 주목을 받고 있다.

### 2.1 의존 문법

한국어 분석에 사용되는 의존 문법에서는 분석의 효율성을 위해 지배소 후위 원칙 및 투영성(projectivity) 원리를 제약사항으로 적용한다. 한국어는 문장에서 의존소 뒤에 지배소가 온다는 원칙을 가정하는 것이 일반적이다[1]. 형태소 단위의 품사태그 기반 의존문법은 한국어의 PSMC 현상을 다룰 수 있는 반면, 구성체의 상태를 표현하는 실질형태소와 문장 내에서의 구문적 역할을 나타내는 형식형태소를 구별하지 않는다. 또한, 기존 의존 문법에서는 첨용에 의해 발생하는 의존 관계와 어절간에 일어나는 구문적인 의존 관계를 상호 구분하지 않는다.

한편, 어절 단위의 품사태그 기반 의존문법에서는 어절이 구문 분석을 위한 기본 단위가 되는데, 한 어절은 다른 어절로부터 수식을 받거나 또는 다른 어절을 수식할 수 있기 때문에, 각 어절의 레이블은 (ltag, rtag) 형태를 취하게 된다. 한국어의 지배소 후위 원칙에 의하면, ltag는 앞에 선행하는 의존소(어절)에 대하여 본 어절이 가지는 지배소로서의 품사태그를 나타내며, rtag는 후위 지배소(어절)에 대해 본 어절이 가지는 의존소로서의 품사태그를 나타내게 된다. 문장 (1)을 어절 품사태그 기반 의존문법에 맞게 작성한 코퍼스는 표 1과 같다.

표 1 어절 품사태그 기반 의존문법의 표현

| 어절 번호 | 지배소 어절 | 어절 내용 | 어절 레이블 |
|---|---|---|---|
| 1 | 4 | 시간/ncn+이/jcs | (ncn, jcs) |
| 2 | 3 | 귀중/ncps+하/xsm+ㄴ/etm | (paa, etm) |
| 3 | 4 | 것/nbn | (nbn, nbn) |
| 4 | 0 | 이/jp+다/ef+./sf | (jp, sf) |

여기에서 "것+이+다+."는 인위적인 방법에 의해 두 어절로 분리되어 있다. 이와 같이 "것+이+다+."의 어절을 2개의 다른 어절로 분리하여 처리한 이유는 "것+이+다+."가 하나의 어절 상태로는 "시간이"와 "귀중한"의 두 어절의 수식 관계를 동시에 만족시킬 수 없기 때문이다.

### 2.2 구구조 문법

구구조문법은 규칙의 오른편(right hand side: RHS)에 나오는 단말기호 및 비단말기호의 개수에 대한 제한이 없고 체계적인 구절 생성 방법을 제공하지 않으므로, 이러한 특성은 곧 문법 규칙의 작성시 정형성의 미흡이라는 현상으로 나타나게 된다[8]. 또한, 확률 구구조문법에서는 RHS 길이의 변이로 인해 문장에 대한 트리 구성 참여 규칙 수가 가변적이 되며, 작은 트리가 큰 트리

에 비해 선호도를 가지게 된다.

제한된 구구조문법(RPSG)은 한국어의 자유 어순 특성을 다루기 위해 제안된 문법이다[3]. 제한된 구구조문법에서는 규칙을 세 가지 형태로 제한한다. 형태 1 규칙 ($A \rightarrow B+\tau$)은 파생 현상을 기술하기 위한 것이고, 형태 2 규칙($A \rightarrow B+\gamma\ C$)은 구문요소(constituent) 사이의 관계를 표현하기 위한 규칙이며, 형태 3 규칙 ($A \rightarrow A1 + \gamma\ A2+\gamma \ldots\ An$)은 병렬을 기술하기 위한 규칙이다.

문장 (1)을 Penn Korean Treebank[4]의 구구조문법 형식과 KAIST 코퍼스[9]에서 사용한 제한된 구구조문법 형식으로 나타내면 그림 1에서 보는 바와 같다.

구구조문법과 제한된 구구조문법은 품사태그와 관계없이 인위적으로 정의한 소수개의 구문 레이블 - 한국어의 경우 [3]에서는 8개 레이블, [4]에서는 11개 레이블 - 을 사용한다. 그러나 품사태그 자체는 구문 정보뿐만 아니라 형태결합 정보와 의미정보 등도 같이 담고 있기 때문에[10], 이렇게 인위적이고 한정된 구문 레이블을 사용함으로써 품사태그에 내재되어 있는 정보를 잃어버리는 현상을 초래하게 된다.

명사구 "귀중+하+ㄴ 것"에서 구구조문법은 PSMC 현상을 제대로 다루지 못하고 있는 반면, 제한된 구구조문법에서는 형태 2 규칙 "*NP → ADJP+etm NP*"를 사용하여 처리하고 있다. 또한, 구구조문법은 "귀중+하+ㄴ 것+이" 구성체를 표현하는 데 있어 명사구가 동사구로 변환되는 과정을 보여주지 못하는 반면, 제한된 구구조문법에서는 형태 1 규칙 "*VP → NP+jp*"를 이용하여 이를 처리한다.

한편, 구구조문법이 주격 명사구 "시간+이"를 "*NP-SBJ*"로 표현하는 반면에 제한된 구구조문법에는 이에 상응하는 레이블이 없다. 제한된 구구조문법은 형식형태소를 실질형태소 사이의 연결점으로만 취급하기 때문에, 어순 교차의 단위(unit of scrambling)를 표현하지 못할 뿐만 아니라 그것이 문장 내에서 가져야 할 주제적 역할도 부여할 방법이 없다. 이는 구성적 언어(configurational language)에서 정한 구문 레이블들을 그대로 사용하다 보니, 그 자체가 문장성분이 되는 해당 언어에서와는 달리 한국어에서는 첨용 및 활용 접미사가 붙어 문장성분이 되는 특성을 무시한 데 기인한다. 그 결과 "*S → VP+ef+sf*"와 같은 경우를 제외한 다른 구문 레이블들은 항상 실질형태소로 끝나는 현상이 나타나는 것이다.

## 3. 한국어 구문 분석을 위한 복합 레이블링

일반적으로 규칙에서 RHS의 길이가 가변적이면 해당 RHS를 포괄하는 규칙의 개수를 명확하게 결정하기 어렵다. 다시 말해 구구조문법에서 규칙1 ( *NP → ADJP NP*), 규칙2 ( *NP → NP NP*), 규칙3 ( *NP → ADJ NP NP*)이라는 세 가지 규칙이 있을 때 입력이 "아름다운 정원의 꽃"이라면, {규칙1 ⇒ 규칙2} 또는 {규칙2 ⇒ 규칙1} 또는 {규칙3}의 적용이 가능하다. 여기에서 규칙3은 구성요소간 의존 관계가 명시적으로 드러나지 않으며, 규칙1 및 규칙2를 포함하는 잉여 규칙이 된다. 또한 가변 길이의 RHS를 가지게 되면 동일한 입력문에 대해 적용되는 규칙의 수가 가변적이 된다. 이는 확률 파싱에서 각 규칙의 확률에 큰 변이가 없을 경우 적은 규칙으로 이루어지는 구문 트리 결과를 선호하게 되는, 확률 적용의 문제를 유발할 수 있다.

따라서 left-factoring[11]에 의한 이진화 방법(binarization)을 적용한 구구조문법들[12,13]이 제시되어 있다. 그럼에도 불구하고 구구조문법은 비단말기호들을 RHS에 배열하기 때문에 동일 입력문에 대하여 이진화된 규칙이라고 하더라도 적용 규칙의 수가 가변적이 될 수 있다. 따라서 확률 적용상의 편향성(bias)이 완전히 해결되지는 않는다. 이에 비해 의존문법 등 이진 규칙을 적용하는 문법은 동일 입력문에 대하여 적용되는 규칙의 수가 일정하기 때문에 확률 적용상의 편향성이 사라지는 장점이 있다.

복합 레이블이라는 용어는 $C \rightarrow L\ R$ 형태의 이진 분기 규칙에서의 노드 레이블 $C$가 하위 구성체의 레이블인 $L$과 $R$로부터 뽑은 정보로 구성되기 때문에 붙여진 것이다. 여기에서는 이진 분기 규칙의 형식과 복합 레이블을 자동적으로 생성하는 방법에 대하여 기술한다.

### 3.1 이진 규칙

(*S* (*NP-SBJ* 시간/NNC+이/PCA)
(*VP* (*NP* (*ADJP* 귀중/NNC+하/XSJ+ㄴ/EAN)
것/NNX+이/CO+다/EFN))
./SFN)

(a) 구구조문법

(*S* (*VP* (*NP* 시간/ncn)+이/jcs
(*VP* (*NP* (*ADJP* (*NP* 귀중/ncps)+하/xsm)+ㄴ/etm
(*NP* 것/nbn))
+이/jp))
+다/ef+./sf)

(b) 제한된 구구조문법

그림 1 문장 (1)에 대한 구구조문법과 제한된 구구조문법 표현

표 2 각 규칙 형태별 예시

| 형태 | 예 | 규칙 |
|---|---|---|
| 1 | 시간/ncn | *0Ncn* → *ncn* |
| 2a | 실/xp+(*0Ncn* 시간/ncn) | *1Ncn* → *xp* + *0Ncn* |
| 2b | (*0Ncn* 시간/ncn)+들/xsn | *1Ncn* → *0Ncn* + *xsn* |
| | (*0Ncn* 시간/ncn)+이/jcs | *jcsNcn* → *0Ncn* + *jcs* |
| 2c | (*0Ncpa* 근무/ncpa)+(*0Ncn* 시간/ncn) | *NcpaNcn* → *0Ncpa* + *0Ncn* |
| 3 | (*etmPaa* 귀중한) (*0Nbn* 것/nbu) | *EtmNbn* → *etmPaa 0Nbn* |

문맥 자유 문법은 $(T, N, G, S)$의 4개 항으로 이루어지며, $T$와 $N$은 각각 단말기호 (즉, 품사태그) 집합과 비단말기호(즉, 구문 레이블) 집합이고, $S \in N$는 시작기호, $G$는 유한개의 규칙을 가진 집합을 표시한다. 여기에서 $T$는 55개의 품사태그로 이루어진 KAIST 품사태그 집합이다[10].

각 규칙은 $C \rightarrow L\ R$의 이진 형태로서, 왼편(left hand side: LHS) 레이블인 C는 의존 관계 정보와 구문 특성을 동시에 표현토록 한다. 기본적으로 한국어의 경우 RHS에서 L은 의존소이며 R은 지배소이다. 또한, 실질형태소(문장부호 포함)에는 최초 레이블을 부여하는 반면 형식형태소 자체에는 레이블을 부여하지 않는다.[1] 노드 레이블 $C$가 RHS의 지배소 정보, 현 구성체의 상태 정보(구구조문법에서의 *NP, VP* 등), 그리고 현 구성체의 역할 정보(주격, 목적격 등)를 동시에 가지기 위해서는 $C_LC_R$ 형태의 복합 레이블을 가져야 할 필요가 있다. 결과적으로, 규칙 $C \rightarrow L\ R$은 $C_LC_R \rightarrow L_LL_R\ R_LR_R$과 같이 변환된다. 여기에 한국어에서 가지고 있는 형태 결합 표시('+') 및 띄어쓰기 표시(' ')를 고려하면, 다음과 같은 세 가지 형태의 규칙을 정의할 수 있다.

- Type 1 (단일 원소 규칙): $C_LC_R \rightarrow f$, $C_LC_R \in N$, $f \in T$, $f$는 실질 형태소
- Type 2 (어절 내부 규칙): 이 형태에는 어절 내부에서 일어나는 모든 형태의 규칙이 포함된다. 다음과 같은 세 가지 하부 규칙으로 분류된다.
  - Type 2a: $C_LC_R \rightarrow e{+}R_LR_R$
  - Type 2b: $C_LC_R \rightarrow L_LL_R{+}e$
  - Type 2c: $C_LC_R \rightarrow L_LL_R{+}R_LR_R$

  $C_LC_R,\ L_LL_R,\ R_LR_R \in N$, $e \in T$, $e$는 형식형태소
- Type 3 (어절 사이 규칙): $C_LC_R \rightarrow L_LL_R,\ R_LR_R,\ C_LC_R,\ L_LL_R,\ R_LR_R \in N$

표 2는 각 형태의 규칙에 대한 예를 보여주고 있다.

### 3.2 복합 레이블의 자동 생성

이 절에서는 이진 규칙을 위한 복합 레이블 생성 방법을 기술한다. 적절한 복합 레이블을 생성하기 위해 필요한 제약사항은 다음과 같다. 복합 레이블의 어떠한 하위 구성체의 레이블도 null이 아니어야 하고, $C_L$ 및 $C_R$은 중복을 피하기 위하여 동일한 정보를 가지지 않도록 해야 하며, 하위 구성체의 레이블로부터 현 구성체의 레이블을 만드는 방법에는 일관성이 있어야 한다. 또한, $C_L$ 및 $C_R$은 품사태그 정보를 적용하여야 한다.

설명을 위해서 먼저 $L$을 왼편의 하위 구성체, $R$을 오른편 하위 구성체라고 하자. 아래 첨자 $_{State}$는 구성체의 상태 태그를 의미하며, $_{Role}$은 해당 문법 기능을 가진 하위 구성체가 의존소로 결합되어 현 구성체를 이루고 있다는 의미이고, $_{role}$은 해당 문법 기능을 가진 하위 구성체가 지배소로서 현 구성체를 이루고 있다는 의미이다. 다시 말하여, $_{role}$은 현 구성체의 형식형태소가 가지는 역할을 나타내는데, 이 형식형태소는 다른 구성체와 결합하여 상위 구성체를 만들게 된다. 이와 달리 $_{Role}$은 그렇게 결합하여 상위 구성체가 형성되었을 때 상위 구성체의 복합 레이블의 한 구성 레이블을 나타내게 되는 것이다. 예를 들어, "귀중하+ㄴ 것"에 대한 트리를 형성하는 과정을 표현하면 그림 2에서 보는 바와 같다.

'ㄴ/etm'은 관형형어미로 먼저 서술어 "귀중하"에 결합하여 관형어를 만들게 되는데, 이를 표현하는 구문 레이블은 "*etmPaa*"로서 "*etm**"은 서술어의 구문상에서의 역할이 관형어임을 의미하며, 관형어의 수식 기능이 아직 적용되지 않은 상태임을 의미한다. 이와 달리,

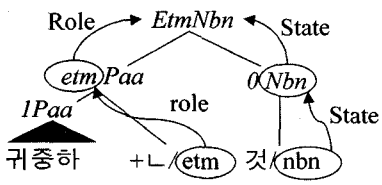



그림 2 상태/역할정보를 적용한 레이블

1) 실질형태소는 그 자체로 독립적인 구문 요소가 될 수 있는 반면, 형식형태소는 실질어와 결합하지 않고는 구문요소로 존재하지 못한다. 문장부호는 형태소 수준이 아니라 문장 수준의 기능을 가지므로, 독립된 단위로 구별하여 문장 차원에서 다루어야 한다.

"*Etm**"은 관형어 "*etmPaa*"가 불완전명사 "것/nbn"과 결합하였음을, 즉 관형어가 한 번 의존관계를 적용시킨 결과임을 나타내는 표현이다. 결과적으로 "귀중한 것"은 관형어 "귀중한"의 수식을 받은 명사구 "것"임을 "*Etm-Nbn*"이라는 복합 레이블을 통하여 표현하게 된다.

한국어의 경우 이러한 상태와 역할은 품사태그에 따라 비교적 쉽게 구분된다. 각 형태소마다 표현하는 상태 또는 역할은 표 3에서 보는 바와 같다. 상태를 나타내는 것들은 실질형태소와 동일하며, 단지 "*Jp*"가 이에 속한다. 상태의 표현은 대문자로 시작하도록 하였다. 역할은 소문자로 표현하였다. 예를 들어, "*jcs**"는 역할을 나타내지만 "*Jcs**"의 경우에는 그렇지 않다.

표 3 상태 및 역할 표시자

| 표시자 | |
|---|---|
| 상태 | *Ncpa, Ncps, Ncn, Nq, Nbu , Nbn, Npp, Npd, Nnc, Nno, F, Ii , Pvd, Pvg, Pad, Paa, Px, Mmd, Mma, Mad, Maj, Mag, Jp* |
| 역할 | *jcs, jco, jcc, jcm, jcv, jca, jcj, jct, jcr, jxc, jxt, jxf, ep, ecc, ecs, ecx, etn, etm, ef* |

구성체의 복합 레이블을 표현하기 위해서 역할에는 소문자를, 상태에는 대문자를 시작 문자로 표현하므로, "*jcsNcn*"의 경우 상태는 "**Ncn*", 역할은 "*jcs**"가 된다. 그러나 "*EtmNbn*"의 경우에는 상태로서 "**Nbn*"을 나타내고 있으나 역할에 대한 정보는 없다. 이는 대문자로 시작하는 "*Etm**"은 현 구성체의 역할을 나타내는 것이 아니고 구성체의 구성 이력을 나타내기 때문이다. 다시 말해, 이 구성체의 복합 레이블은 관형구 "*etm**"에 의해 의존명사(구) "**Nbn*"이 수식을 받음으로써 전체적으로 관형어의 수식을 받는 의존명사구임을 나타낸다. 이러한 방식을 적용하여 각 형태의 규칙마다 복합 레이블을 생성하는 방법은 다음과 같다.

- **Case 0** (하위 구성체가 한 개의 상태만을 가지는 경우): type 1, 접두사를 가지는 type 2a, 파생접미사를 가지는 type 2b 규칙들이 속한다. Type 1 규칙에는 "0"을, type 2 규칙에는 "1"을 사용한다.
- **Case 1** (*L*이 상태만을 가지고 *R*이 역할만을 가지는 경우): type 2b (e가 'jp'인 경우는 제외)가 여기에 속한다. 복합 레이블은 $R_{role}L_{State}$으로서, 결과적으로 생성된 구성체의 역할은 $R_{role}$이고 상태는 $L_{State}$가 된다.
- **Case 2** (*L*과 *R* 모두 상태만을 가지는 경우): type 2b (*e*가 'jp'인 경우)와 type 2c 규칙이 이에 속한다. 복합 레이블 $L_{State}R_{State}$는 결과적으로 생성되는 구성체가 역할 정보를 가지지 않고 상태로서 $R_{State}$를 가지고 있음을 나타낸다.
- **Case 3** (*L*은 상태와 역할을 가지고 있으나 *R*은 상태만을 가지는 경우): type 3 규칙들이 이에 속한다. 결과적으로 생성된 구성체는 역할을 가지지 않는다.
- **Case 4** (*L*과 *R* 모두 상태 및 역할을 가지는 경우): type 3 규칙의 예외적인 현상

| Case 0 | | 예 | 상태 | 역할 | LHS 레이블 |
|---|---|---|---|---|---|
| Type 1 | L | | x | x | $0R_{State}$ |
| | R | 시간/ncn | O(Ncn) | x | *(0Ncn)* |
| | rule | *0Ncn → ncn* | | | |
| Type 2a | L | 실/xp | x | x | $1R_{State}$ |
| | R | +*(0Ncn* 시간/ncn) | O(Ncn) | x | *(1Ncn)* |
| | rule | *1Ncn → xp + 0Ncn* | | | |
| Type 2b | L | (*0Ncn* 시간/ncn) | O(Ncn) | x | $1L_{State}$ |
| | R | +들/xsn | x | x | *(1Ncn)* |
| | rule | *1Ncn → 0Ncn + xsn* | | | |

| Case 1 | | 예 | 상태 | 역할 | LHS 레이블 |
|---|---|---|---|---|---|
| Type 2b | L | (*0Ncn* 시간/ncn) | O(Ncn) | x | $R_{role}L_{State}$ |
| | R | 이/jcs | x | O(jcs) | (*jcsNcn*) |
| | rule | *jcsNcn → 0Ncn + jcs* | | | |

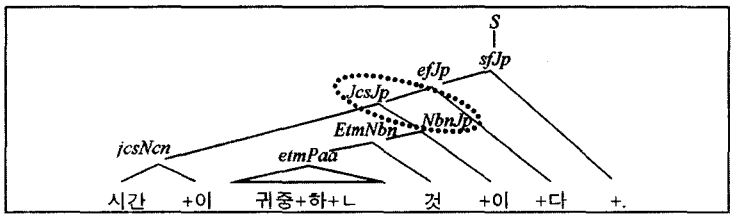
| Case 2 | | 예 | 상태 | 역할 | LHS 레이블 |
|---|---|---|---|---|---|
| Type 2b | L | (*EtmNbn* 귀중+하+ㄴ 것) | O(Nbn) | x | $L_{State}R_{State}$ |
| | R | +이/jp | O(Jp) | x | (*NbnJp*) |
| | rule | *NbnJp → EtmNbn + jp* | | | |
| Type 2c | L | (*0Ncpa* 근무/ncpa) | O(Ncpa) | x | $L_{State}R_{State}$ |
| | R | +(*0Ncn* 시간/ncn) | O(Ncn) | x | (*NcpaNcn*) |
| | rule | *NcpaNcn → 0Ncpa + 0Ncn* | | | |

| Case 3 | | 예 | 상태 | 역할 | LHS 레이블 |
|---|---|---|---|---|---|
| Type 3 | L | (*etmPaa* 귀중+하+ㄴ) | O(Paa) | O(etm) | $L_{Role}R_{State}$ |
| | R | (*0Nbn* 것/nbn) | O(Nbn) | x | *(EtmNbn)* |
| | rule | *EtmNbn → etmPaa 0Nbn* | | | |

| Case 4 | | 예 | 상태 | 역할 | LHS 레이블 |
|---|---|---|---|---|---|
| Type 3 | L | (*jcaNbn* 9+시+부터) | O(Nbn) | O(jca) | $R_{role}R_{State}$ |
| | R | (*jcaNbn* 6+시+까지) | O(Nbn) | O(jca) | *(jcaNbn)* |
| | rule | *jcaNbn → jcaNbn jcaNbn* | | | |

(*S* (*sfJp* (*efJp* (*JcsJp* (*jcsNcn* (*0Ncn* 시간/ncn) + 이/jcs)
(*NbnJp* (*EtmNbn* (*etmPaa* (*1Paa* (*0Ncps* 귀중/ncps) + 하/xsm) + ㄴ/etm)
(*0Nbn* 것/nbn)) + 이/jp)) + 다/ef) + ./sf))

(a) 문장 (1)에 대한 괄호 표현




(b) 문장 (1)에 대한 트리 구조 표현

그림 3 제안한 규칙 기술 방법에 따른 문장 (1)의 표현

이진 규칙 레이블을 위한 상태 및 역할의 가능한 조합은 36가지({$L_{State}$, $L_{Role}$, $L_{role}$} × {$R_{State}$, $R_{Role}$, $R_{role}$} × {*LL, LR, RL, RR*})가 된다. 하지만, 위에서 나타나는 것처럼 유효한 조합은 5가지뿐인데, 그 이유는 다음과 같다.

1. *LL* 및 *RR* 조합은 발생하지 않는다. 단, case 4는 예외.
2. $L_{Role}R_{Role}$과 같이 상태 정보 없이 역할과 역할만의 조합을 이루는 것은 무의미하다.
3. 레이블 구성시 역할 정보를 나타내는 부분이 항상 상태 정보를 나타내는 부분보다 먼저 오도록 표현함으로써 표현의 일관성을 가지도록 한다.
4. $L_{role}R_{State}$와 $R_{Role}L_{State}$는 구성체를 표현할 수 없다.

"시간/ncn+이/jcs"에 대한 case 1 규칙 "*jcsNcn → 0Ncn+jcs*"에 대한 설명은 다음과 같다. 먼저, '시간/ncn'은 실질형태소이므로 단일 원소 규칙인 "*0Ncn → ncn*"이 적용된다. 이후 뒤에 나오는 '이/jcs'에 대해 의존소로서 결합한다. '이/jcs'는 구성체에서 주격 역할인 "*jcs**"를 취하며, 복합 레이블은 주격 명사구라는 표현의 "*jcsNcn*"이 된다. 여기에서 소문자 "*jcs**"는 주격조사가 어순상 명사(구) 뒤에 나온다는 표현이다. 그림 3은 문장 (1)을 이러한 복합 레이블을 이용하여 트리 구조로 표현한 것이다. "*JcsJp → jcsNcn NbnJp*"는 주격명사구 "*jcsNcn*"이 의존소로서 지배소인 서술어구 "*NbnJp*"와 결합하여, 결과적으로 주어의 수식을 받는 서술어 구성체 "*JcsJp*"가 됨을 의미한다.

## 4. 실 험

이 절에서는 제안한 규칙 기술 방법이 그 자체로 한국어 구문 분석에 적합한지를 판단하기 위한 실험 결과를 기술한다. KAIST 구문 트리 코퍼스는 55개의 KAIST 품사태그와 제한된 구구조문법을 적용한 31,080 문장으로 구성되어 있다[9]. 문장당 평균 어절 수 및 형태소 수는 각각 11.35 및 25.62이다.

실험을 위해서 KAIST 코퍼스를 제안한 규칙 기술 방법에 맞게 변환하는 작업 후, 교차 시험 (cross validation: CV)[2]을 실시하였다. 실험은 PC상의 Hancom Linux 2.2 환경에서 MSLR 파서[14]를 이용하여 수행하였으며, 결과 측정을 위해 PARSEVAL 평가 기준[3][15]

2) 교차 시험이란 코퍼스를 일정 비율로 균등 분할하여 한 주기가 될 때까지 각 부분을 돌아가면서 시험하는 것으로, 본 논문에서는 전체 코퍼스를 10등분하여 각 부분에 대한 held-out test-학습에 사용되지 않은 문장들에 대하여 시험하는 것으로 9개 부분을 학습시켜 나머지 1개 부분을 시험- 를 실시하였다.

3) 정확율(precision)은 분석 결과에 포함된 올바른 구문요소(constituent) 개수의 비율, 재현율(recall)은 올바른 구문 분석에 포함된 구문요소 개수 중 분석 결과에 포함된 개수의 비율이다. 유표지 정확율(labelled precision:LP) 및 유표지 재현율(labelled recall: LR)은 구문요소의 맞고 틀림을 그 구문요소의 대상범위와 레이블의 일치 여부까지 판단하여 결정하는 것이며, 무표지 정확율(unlabelled precision)

표 4 KAIST 코퍼스에 대한 문법 규칙 기술 방법의 실험 결과

| 실험 | 문법 | 실험문장 | 레이블 | 학습 | 애매성 | LP | LR | 집단 LP&LR |
|---|---|---|---|---|---|---|---|---|
| 1 | PSG | 1,000 | 8 | 5,262 | 8.71E+08 | 74.66 | 72.18 | |
| 2 | RPSG-1 | 1,000 | 8 | 2,654 | 5.00E+08 | 74.09 | 72.36 | |
| 3 | RPSG-2 | 31,080CV | 8 | 2,381 | 3.76E+14 | 75.39 | 72.83 | |
| 4 | 55태그 | 31,080CV | 726 | 10,735 | 4.69E+06 | 75.58 | 75.58 | 79.30 |
| 5 | 35태그 | 31,080CV | 343 | 5,868 | 1.34E+07 | 73.05 | 73.05 | 76.73 |
| 6 | MP-based DG | 31,080CV | 55 | 1,027 | 7.97E+07 | 75.32 | 75.32 | |

을 적용한 EVALB 프로그램[16]을 이용하였다. 표 4는 이러한 환경에서 실행된 실험 결과를 보여주고 있다.

실험 1과 2는 [3]에서 제시된 것이며, 실험 3은 같은 실험 환경에서 교차 시험한 결과이다. 구구조문법과 제한된 구구조문법에서의 구문 레이블의 개수는 8개이다. 위 세 가지 실험 결과를 종합해 보면 한국어에 대해서 약 75%정확도를 보이고 있는데, 이는 영어의 경우 문맥 자유 문법에 의한 확률 파싱이 75% 전후의 정확도를 낸다[17]는 사실과 유사한 결과이다.

나머지 실험들은 PC상에서 HanCom Linux 2.2와 MSLR 파서를 사용하여 수행되었다. 실험 4는 제안한 규칙 기술 방법에 의한 실험으로서, 55 개의 품사태그를 적용하였으며, 코퍼스로부터 추출된 구문 레이블의 개수는 726개이다. 구문 레이블은 $C_LC_R$로 나타낼 수 있으며 $C_R$은 항상 상태를 표현하므로, $C_R$로 가능한 것은 31개 품사태그들이다[4]. 동일한 방법으로 $C_L$로 쓰일 수 있는 개수는 68개가 된다. 따라서, 2,108개의 구문 레이블이 가능한데, 형태소간 결합 제약 및 구문적인 결합 제약으로 인하여 코퍼스에서 726개의 레이블만이 발견된 것이다. 또한, 그중 잘못된 구문 레이블이 105개 발견되었지만, 실험에서는 잘못된 오류 레이블이라고 하더라도 코퍼스에서 자동으로 추출된 경우에는 그대로 포함하여 수행하였다. 실험 4의 결과를 보면 LP 및 LR 측정 지표를 적용한 정확율이 75.58%로서, 기존 방법들과 비교할 때 구문 분석 규칙의 기술 방법으로서 문제가 없음을 알 수 있다.

실험 5는 실험 4와 비교하여 간략화된 태그 집합이 미치는 영향을 평가하기 위해 35개의 품사태그를 적용한 결과이다. 실험 4에 사용된 55개 품사태그를 수준 2에서 묶어 (예: pvg, pvd를 pv로) 35개 품사태그로 만들어 실험한 것으로서, 55개 품사태그를 그대로 이용한 실험에 비해 2.53%의 성능 저하를 나타냈다. 그 이유는 품사태그 개수를 줄이면서 품사태그에 포함된 정보의 변별력이 저하되었기 때문으로 풀이된다. 실험 6은 형태소 품사태그 기반 의존문법의 경우이다. 형태소 품사태그 기반 의존문법은 각 구성체가 지배소의 품사태그에 의해 표현되기 때문에 55개의 레이블을 가지는데, 정확율 및 재현율을 보면 75%보다 약간 상회하는 것으로 나타나 있다.

실험 4와 5의 결과 중, "집단 LP&LR"은 유표지 정확율과 유표지 재현율을 평가하는 데 있어서, 726개의 구문레이블을 각각 55개와 35개로 집단화시켜 평가해 본 것이다. 이와 같이 구문레이블을 집단화시켜 평가해 본 이유는 다음과 같다.

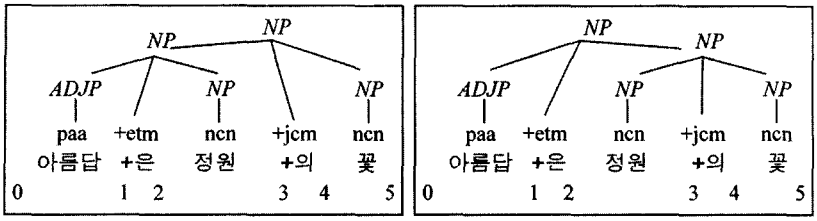
그림 4는 문장 "아름다운 정원의 꽃"에 대한 구문 트리 표현들이다. (a1)과 (b1)트리가 정답트리이고, (a2)와 (b2)가 구문분석기의 결과트리이다. 그림에서 보는 바와 같이, (a1)과 (b1)은 동일한 구문 수식 정보를 표현하고 있으며, 마찬가지로 (a2)와 (b2)또한 동일한 구문 수식 정보를 나타내고 있다. 그런데, 표 5에서 보는 바와 같이 유표지 정확률과 재현율을 고려할 때, 트리 (a)의 경우에는 NP[2,5]만이 틀린 구조로 평가받는데 비해, 트리 (b)의 경우에는 모두 틀린 구조로 간주된다.

구구조문법 및 제한된 구구조문법에서는 8개의 비단말기호를 사용하므로, 정확율 및 재현율은 8개의 레이블에 대해서만 측정된다. 이러한 측면에서 제안된 규칙 기술 방법을 좀더 공정하게 평가하기 위한 집단 정확율(clustered LP) 및 집단 재현율(clustered LR)을 제시할 수 있는데, 예를 들어 구문 레이블 "*EtmNcn*"은 비복합적인(non-composite) 비단말기호 "*Ncn*"으로 사상될 수 있다. 이렇게 하면 726개의 복합 레이블은 실험 4의 경우 55개 또는 실험 5의 경우 35개의 비단말기호(품사태그의 개수와 같음)로 사상될 수 있다. 이러한 사상을 통한 평가 방법에서는 출력 트리상의 "*EtmNcn*[0,5]"과 정답 트리 "*JcmNcn*[0,5]" 모두 동일한 "*Ncn*[0,5]"으로 판정된다. 실험 4에서 집단 LP & LR의 방법으로 평가했을 경우, 제안한 규칙 기술 방법이 79.3%의 정확률로 기존의 다른 방법에 비해 4% 이상의 향상된 결과를 보였다.

---

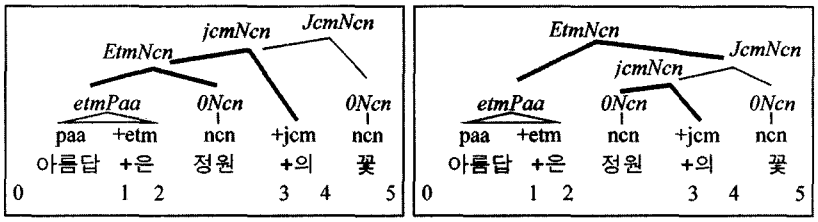
및 무표지 재현율(unlabelled recall)은 레이블의 일치 여부를 고려하지 않은 것이다.

4) KAIST-POS의 문장부호(symbol), 체언(nominal), 외국어(foreign), 감(interjection), 서술어(predicate), 수식어(modifier), 서술격조사(co 이에 포함된다.




(1) a flower in the beautiful garden (2) a beautiful flower in the garden

(a) 제한된 구구조문법의 트리 구조 표현




(1) a flower in the beautiful garden (2) a beautiful flower in the garden

(b) 제안된 규칙 기술 방법에 따른 트리 구조 표현

그림 4 "아름다운 정원의 꽃"에 대한 구문 트리 표현들

표 5 각 구문 규칙 기술방법에 따른 LP 평가(규칙 적용범위를 [] 안에 표현)

| (a1): 정답 트리 | (a2): 결과 트리 | 평가 |
|---|---|---|
| *NP*[0,3] → *ADJP*[0,1]+*etm NP*[2,3] | *NP*[2,5] → *NP*[2,3]+*jcm NP*[4,5] | X |
| *NP*[0,5] → *NP*[0,3]+*jcm NP*[4,5] | *NP*[0,5] → *ADJP*[0,1]+*etm NP*[2.5] | O |
| (b1): 정답 트리 | (b2): 결과 트리 | 평가 |
| *EtmNcn*[0,3] → *etmPaa*[0,2] *0Ncn*[2,3] | *jcmNcn*[2,4] → *0Ncn*[2,3]+*jcm*[3,4] | X |
| *jcmNcn*[0,4] → *EtmNcn*[0,3]+*jcm*[3,4] | *JcmNcn*[2,5] → *jcmNcn*[2,4] *0Ncn*[4,5] | X |
| *JcmNcn*[0,5] → *jcmNcn*[0,4] *0Ncn*[4,5] | *EtmNcn*[0,5] → *etmPaa*[0,2] *JcmNcn*[2,5] | X |

KAIST 품사 태그 집합

| morpheme | | tag |
|---|---|---|
| 실질 형태소 | Symbol | sp(pause), sf(full stop), sl(left quotation & parenthesis mark), sr(right quotation & parenthesis mark), sd(dash), se(ellipsis), su(unit), sy(other symbols) |
| | Nominal | ncpa(active-predicative common noun), ncps(stative-predicative common noun), ncn(non-predicative common noun), nq(proper noun), nbu(unit bound noun), nbn(non-unit bound noun), npp(personal pronoun), npd(demonstrative pronoun), nnc(cardinal numerals), nno(ordinal numerals) |
| | Foreign | f(foreign word) |
| | Intj | ii(interjection) |
| | Predicate | pvd(demonstrative verb), pvg(general verb), pad(demonstrative adjective), paa(attributetive adjective), px(auxiliary verb) |
| | Modifier | mmd(demonstrative adnoun), mma(attributive adnoun), mad(demonstrative adverb), maj(conjunctive adverb), mag(general adverb) |
| 형식 형태소 | Josa | jcs(subjective), jco(objective), jcc(complemental), jcm(adnominal), jcv(vocative), jca(adverbial), jcj(conjunctive), jct(comitative), jcr(quotative), jxc(common auxiliary), jxt(topical auxiliary), jxf(final auxiliary), jp(predicative case) |
| | Ending | ep(prefinal), ecc(coordinate), ecs(subordinate), ecx(auxiliary conjunctive), etn(nominalizing), etm (adnominalizing), ef(final) |
| | Affix | xp(prefix), xsn(noun-derivational), xsv(verb-derivational), xsm(adjective-derivational), xsa(adverb-derivational) |

구문 레이블링의 사례

| | |
|---|---|
| 0Ncn → ncn [시간] | NcnNbn → etnPvg [먹+기] 0Nbn [때문] |
| 1Ncn → xp [신] + 0Ncn [기술] | NbnJp → EtmNbn [귀중한 것] + jp [이] |
| 1Ncn → 0Ncn [시간] + xsn [들] | JxtJp → jxtNpp [그는] NcnJp [학생이] |
| 1Pvg → 0Ncpa [생각] + xsm [하] | NcpaNcn → spNcpa [정치,] 0Ncn [경제] |
| 1Mag → 0Ncps [영원] + xsa [히] | MagPaa → jxcMag [너무나] 0Paa [길] |
| jcoNcn → etnPvg [오+기] + jco [를] | EccPaa → spJp [좋고,] 0Paa [부지런하] |
| jcsNcn → 0Ncn [시간] + jcs [이] | eccJp → eccJp[역동적이면서] + jxc [도] |
| JcjNcn → jcjNcn [복수+나] etnPvg [붓+기] | spNcpa → 0Ncpa [정치] + 0Sp [,] |
| JcmNcn → jcmNcn [정원의] 0Ncn [꽃] | |

## 5. 결 론

이 논문에서는 한국어의 구문적 특성과 한국어 구문 분석을 위한 기존 문법들을 살펴보고, 새로운 규칙 기술 방법 및 레이블 생성 방법으로 이진 분기 규칙과 품사 태그를 이용한 복합 레이블 자동 생성 방법을 제시하였다. 제안된 규칙에서는 구문 레이블이 구성체의 상태 및 역할을 자연스럽게 표현할 수 있으며, 동시에 의존 관계 및 구문적 정보를 표현한다. 또한, 표현된 구문 레이블을 통해서 해당 구문 트리의 적절성을 쉽게 판별할 수 있다. 형태소 기반의 이진 규칙을 적용함으로써 한국어의 특성인 PSMC 현상을 다룰 수 있으며, 확률적 구문 분석에서 확률 적용의 편향성(bias)을 방지한다.

제안된 규칙 기술 방법은 하위 구성체들의 노드 레이블을 활용하고 품사태그를 사용함으로써 규칙에 코퍼스의 구문 트리 확률을 보다 적절하게 반영한다. 이진 규칙을 이용한, 정규화된 규칙 표현 방법을 통하여 다양한 트리 표현 방법에 의해 생성된 트리들을 쉽게 상호 비교할 수 있다. 실험 결과 75.58%의 LP 및 LR, 79.30%의 집단 정확율 및 재현율을 보이며, 이는 기존 방법에 비해 4%이상 향상된 결과이다. 또한, 제시된 규칙 기술 방법은 품사태그를 바탕으로 한 범용적 레이블링 알고리즘을 이용하여 복합 레이블을 생성하므로, 일본어와 같은 교착어뿐만 아니라 영어와 같은 굴절어에도 적용 가능할 것이다.

## 참 고 문 헌


[1] C. H. Kim, J. H. Kim, J. Y. Seo, and G. C. Kim. 1994. A right-to-left chart parsing with headable paths for Korean dependency grammar. *Computer Processing of Chinese and Oriental Languages* 8 (Supplement), 105~118.

[2] K. J. Seo, K. C. Nam, and K. S. Choi. 1998. A probabilistic model for dependency parsing considering ascending dependencies. *Literary and Linguistic Computing* 13(2), 59~63.

[3] K. J. Lee, J. H. Kim, and G. C. Kim. 1997. An efficient parsing of Korean sentences using restricted phrase structure grammar. *Computer Processing of Oriental Languages* 11(1), 49~62.

[4] C. H. Han, N. R. Han, and E. S. Ko. 2001. *Bracketing Guidelines for Penn Korean TreeBank.* IRCS Report 01-10, University of Pennsylvania.

[5] 나동렬. 1994. 한국어 파싱에 대한 고찰. *정보과학회지* 12(8), 33~46.

[6] J. Cha and Geunbae Lee. Structural disambiguation of morpho-syntactic categorial parsing for Korean, *Proceedings of 18th Conference on Computational Linguistics*, pp 1002~1006. 2000.

[7] Jeongwon Cha, Geunbae Lee, Jong-Hyeok Lee. Korean Combinatory Categorial Grammar and statistical parsing, *Computers and the Humanities*, Vol 36(4): 431~453, Nov. 2002.

[8] C. D. Manning and H. Schutze. 1999. *Foundations of Statistical Language Processing.* The MIT Press.

[9] 한국과학기술원. 1997. *문화체육부와 과학기술부의 연구과제 국어정보처리기 구축과 STEP2000에서 구축된 KAIST 코퍼스*, 1996~1997. 한국과학기술원.

[10] 최기선, 남영준, 김진규, 한영균, 박석문, 김진수, 이춘택, 김덕봉, 김재훈, 최병진. 1996. 한국어정보베이스를 위한 형태·통사 태그 표준에 관한 연구. *인지과학* 7(4), 43~61.

[11] J. E. Hopcraft and J. D. Ullman. 1979. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley.

[12] E. Charniak, S. Goldwater, and M. Johnson. 1998. Edge-based best-first chart parsing. *Proc. of the Fourteenth Nat'l Conf. on AI*, 127~133.

[13] C. D. Manning, and R. Carpenter. 1997. Probabilistic parsing using left corner language models. cmp-lg/9711003.

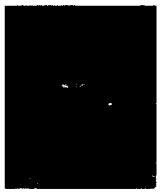
[14] H. Tanaka, T. Tokunaga, and M. Aizawa. 1995. Integration of morphological and syntactic analysis based on LR parsing algorithm. *Journal of Natural Language Processing* 2(2), 59~74.

[15] E.Black, S. Abney, D. Flickinger, C. Gdaniec, R. Grishman, P.Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M.Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syn-

tactic coverage of English grammars. *Proceedings of Speech and Natural Language Workshop*, DARPA, Pacific Grove, 306~311.

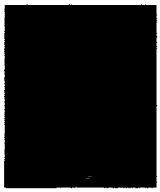
[16] S. Sekine and M. Collins. 1997. Evalb. ftp://cs.nyu.edu/.

[17] C. M. White. 2000. *Rapid Grammar Development and Parsing Constraint Dependency Grammars with Abstract Role Values*. Ph.D. Thesis, Purdue University.



김 성 용

1985년 서울대학교 계산통계학과(학사) 1987년 한국과학기술원 전산학과(공학석사). 2003년 한국과학기술원 전자전산학과 전산학전공(공학박사). 1987년~현재 국방과학연구소 연구원. 관심분야는 자연언어처리, 정보검색, 지능형에이전트



이 공 주

1992년 서강대학교 전자계산학과(학사) 1994년 한국과학기술원 전산학과(공학석사). 1998년 한국과학기술원 전산학과(공학박사). 1998년~2003년 (주)한국마이크로소프트 연구원. 2003년~현재 이화여자대학교 컴퓨터학과 대우전임강사. 관심분야는 자연언어처리, 자연어 인터페이스, 기계번역, 정보검색

최 기 선

1978년 서울대학교 수학과(학사). 1980년 한국과학기술원 전산학과(석사). 1986년 한국과학기술원 전산학과(박사). 1985년~1986년 한국외국어대학교 교수. 1987년~1988년 일본 NEC C&C 초빙연구원 1997년~1998년 미국 스탠포드 CSLI 객원교수. 1988년~현재 한국과학기술원 교수. 1998년~현재 KORTERM 소장. 관심분야는 자연언어처리, 기계번역, 정보검색, 전문용어