다중 파스 여과에 기반한 한국어의 구조적 중의성 해소*

엄미현°, 신대규, 임병준, 나동렬
연세대학교 전산학과

Resolving Structural Ambiguity of Korean Based on Multiple Parse Filtering

Mi-Hyun Eom, Dae-Gyu Shin, Byung-Jun Lim, Dongyul Ra
Dept. of Computer Science, Yonsei Univ.

요 약

본 논문은 한국어의 구문 분석시 생기는 구조적 중의성 현상들 중 명사구를 용언에 부착할 때 발생하는 많은 중의성과 관형용언구가 명사구를 수식할 때 발생하는 중의성들을 해소하는 방법에 대해 다룬다. 대부분의 한국어 문장들이 이러한 중의성 현상을 포함한다. 우리는 문장 분석시 나타나는 다중 파스들을 스택을 이용하여 관리하여 중의성에 의한 다중 파스들을 비교하여 적합하지 않은 파스들을 여과하므로써 중의성을 해소한다. 중의성 해소를 위한 정보원으로서 격 정보와 통계 정보를 이용하는 알고리즘을 제시한다.

1. 서 론

한국어는 다른 자연어와 마찬가지로 많은 중의성을 가지고 있다[7]. 한국어를 분석하는데 있어서 중의성의 해소는 매우 중요한 문제이다. 이 문제를 해결하지 않고 실용적인 한국어 분석 시스템을 개발한다는 것은 불가능하다. 지금까지 한국어의 분석에 관한 중의성에 대해서 많은 연구가 있었으나[1,3,4,5] 아직 만족할 만한 해결책은 나오지 않았다. 본 논문에서는 대부분의 한국어 문장에서 나타나는, 명사구를 용언에 부착할 때 생기는 중의성과 관형용언구가 명사구를 수식할 때 생기는 많은 중의성의 해소를 다룬다. 이러한 중의성은 한국어에서 발생하는 중의성 중에서 가장 많이 발생하는 것으로 생각된다. 격 정보와 통계 정보를 상호 보완적으로 이용하므로서 이러한 중의성을 효과적으로 해소할 수 있었다.

한국어를 표현하기 위해 확장문맥자유문법(ECFG)를 이용하였으며 기존의 챠트 파서를 확장하므로서 이에 대한 파서를 개발할 수 있었다. 문장 분석시에 중의성으로 인하여 많은 다중 파스가 발생하는 데 이들을 통계적인 정보에 의하여 비교 및 제거하므로써 중의성을 해소하도록 하였다.

2. 파서의 개괄

2.1. 한국어의 구문 구조

한국어에 존재하는 구조적인 특성을 이용하며 부분자유 어순의 성질을 처리할 수 있기 위해 우리는 확장 문맥자유문법(ECFG)를 이용한다[6]. 그리고 중심어 후행 성질을 처리하기 위해 역방향 파싱(backward-parsing) 기법을 이용한다. 한국어를 파싱 연구에 이용된 ECFG의 한 예는 다음과 같다.

NA → (Adj | NP$_{[mod]}$ | S$_{[mod]}$)* N* N | S$_{[nf]}$
NB → (NP$_{[cj]}$)* NA | (NA Adv$_{[cj]}$)* NA
NP → NB P
VA → V E | NB DP
VP → VA | VA VA | VA VA VA
Advp → Adv* Adv
S → (NP|Advp)* {S$_{[adv]}$} (NP|Advp)* VP

* 본 연구는 한국과학재단 '94 목적기초 과제(과제번호:94-0100-01-04-3) 의 지원을 받았음.

$S \rightarrow S_{[cj]}^{+}\ S_{[\neg cj]}$

'{ }'는 나타나지 않거나 한 번 나타나는 것을 의미하며, '*'는 0번 이상의 반복을 나타내며, '+'는 1번 이상의 반복을 나타낸다. '[ ]'로 표시된 것은 자질(feature)을 나타내는데, [mod]는 관형형을 나타내고, [nf]는 명사형을 나타내고, [cj]는 병렬접속어미가 있음을 나타낸다. NA는 '먹음'과 같이 어간에 명사형 전성어미가 붙어 만들어지는 $S_{[nf]}$로 이루어지거나 관형상당어구의 수식을 받는 명사로 이루어질 수 있다. NB는 '연필과 지우개'와 같은 명사의 병렬접속을 가능하게 한다. NP는 NB에 조사 P를 붙여 만든 명사구이다. VA는 어간(V)+어미(E)로 이루어지거나 NB에 서술격조사(DP) '이다'를 붙여 만든다. S는 기본적으로 용언 (VP)로 이루어지며, 용언 앞에 명사구(NP)나 부사구(Advp)가 자유롭게 나올 수 있다. $S_{[adv]}$는 부사절을 나타낸다. 마지막 규칙 $S \rightarrow S_{[cj]}^{+}\ S_{[\neg cj]}$는 '~고'나 '~며'와 같은 대등적 연결어미를 갖는 하나 이상의 대등절과 대등절이 아닌 절이 붙어서 하나의 절을 이루는 것을 말한다.

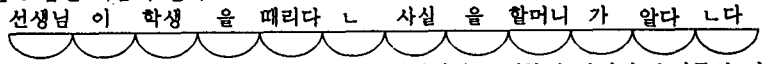
2.2. ECFG를 위한 파서 설계

2.2.1. 차트 파싱을 기반으로 한 파서

ECFG를 이용하여 한국어를 파싱하는 방법을 다음 예를 통해 보자.

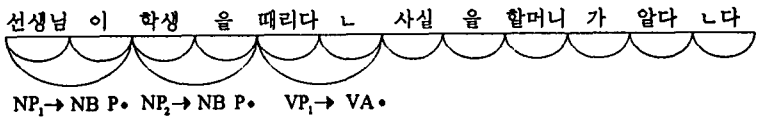
(1) 선생님이 학생을 때린 사실을 할머니가 안다.

(1)의 파싱 전 모습은 다음과 같다.



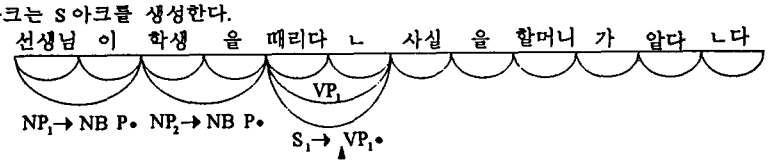
(1)-①

우리는 명사구를 만들어 가며 나아간다. 용언구가 발견되면 그때부터 발견된 용언구의 격을 채울 명사구를 왼쪽으로 나아가며 찾는다. 위의 예에서 '때린'이 발견된 당시의 분석 모습은 다음과 같다.



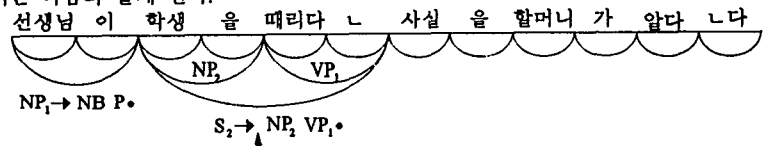
(1)-②

이 때 VP 아크는 S 아크를 생성한다.



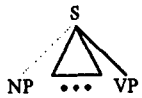
(1)-③

차트의 결합법칙 중에서 NP 아크와 S 아크를 합쳐서 S 아크를 만드는 규칙이 있다(NP-S 규칙). 이 규칙을 적용하면 구문 분석은 다음과 같게 된다.



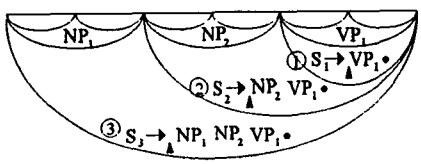
(1)-④

NP-S 규칙의 의미 :



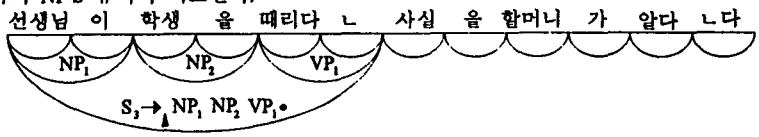
위의 그림에서 실선 부분이 나타내는 S가 있을때 왼편의 NP를 S에 부착하면서 NP가 VP의 격을 채우도록 시도하는 것이다.

역방향 파싱 :



위의 그림에서 파싱의 진행순서는 다음과 같다. NP_1, NP_2, VP_1이 만들어진 후에 ①, ②, ③의 순서로 파싱이 진행된다. VP 아크에 의해 S 아크를 만들고, NP 아크와 S 아크가 인접해 있을 때 챠트를 오른쪽에서 왼쪽으로(역방향으로) 진행하면서 NP 아크를 하나씩 VP 아크에 연결한다. ▲는 역방향 파싱이 되는 것을 의미한다.

(1)-④에 다시 NP-S 규칙이 시도된다.



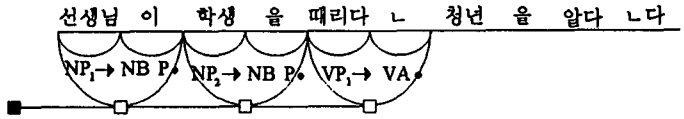
(1)-⑤

2.2.2. 스택을 기반으로 한 다중 파스의 운용

파싱은 스택을 기반으로 이루어진다. 각각의 토큰들이 읽혀져서 스택 안으로 들어오게 되고, 현재까지 읽혀진 입력 스트링에 대한 파스가 스택에 만들어지게 된다. 다음 예를 보자.

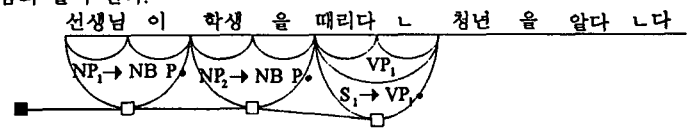
(2) 선생님이 학생을 때린 청년을 안다.

(2)에서 '때린'이 발견된 당시의 스택의 모습은 다음과 같다.



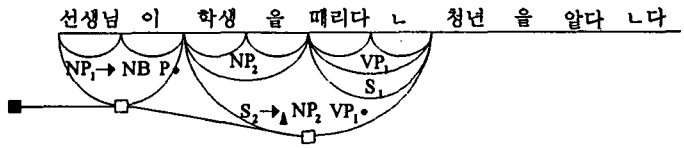
(2)-①

파스는 스택경로 하나로 나타내어진다. 스택경로 하나는 그림에서와 같이 기저(base) ■ 에서 시작되는 인접한 챠트의 나열로 된 한 경로이다(■——□——□ … □) . (2)-①에서 파스는 하나이다. (2)-①에서 VP 아크가 S 아크로 되며 스택은 다음과 같이 된다.



(2)-②

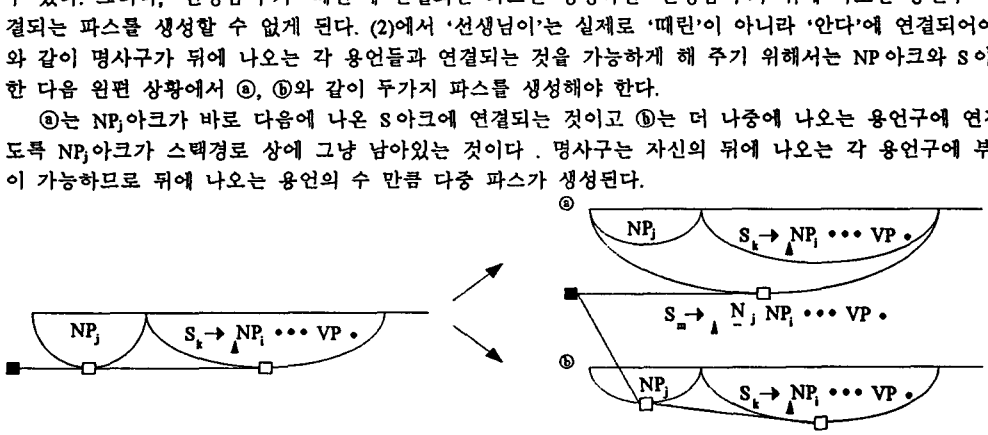
(2)-②에서 파스는 하나이다. (2)-②의 스택은 NP-S 규칙에 의해서 다음과 같이 바뀐다.



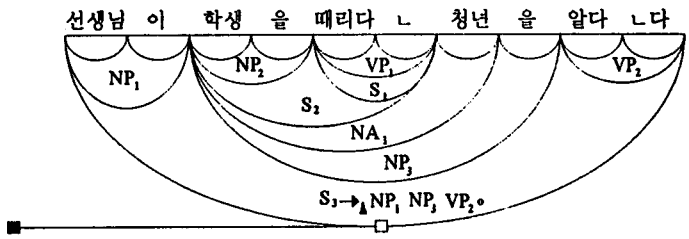
(2)-③

(2)-③에서도 파스는 여전히 하나이다. (2)-③에 NP-S 규칙을 이용하여 (1)-⑤ 처럼 '선생님이'를 '때린'에 연결할 수 있다. 그러나, '선생님이'가 '때린'에 연결되는 파스만 생성하면 '선생님이'가 뒤에 나오는 용언구 '안다'에 연결되는 파스를 생성할 수 없게 된다. (2)에서 '선생님이'는 실제로 '때린'이 아니라 '안다'에 연결되어야 한다. 이와 같이 명사구가 뒤에 나오는 각 용언들과 연결되는 것을 가능하게 해 주기 위해서는 NP 아크와 S 아크가 인접한 다음 원편 상황에서 ⓐ, ⓑ와 같이 두가지 파스를 생성해야 한다.

ⓐ는 NP_j아크가 바로 다음에 나온 S 아크에 연결되는 것이고 ⓑ는 더 나중에 나오는 용언구에 연결될 수 있도록 NP_j아크가 스택경로 상에 그냥 남아있는 것이다 . 명사구는 자신의 뒤에 나오는 각 용언구에 부착되는 것이 가능하므로 뒤에 나오는 용언의 수 만큼 다중 파스가 생성된다.



(2)-③을 ⓑ와 같이 진행하여 문장끝까지 분석했을 때의 파스는 다음과 같다.

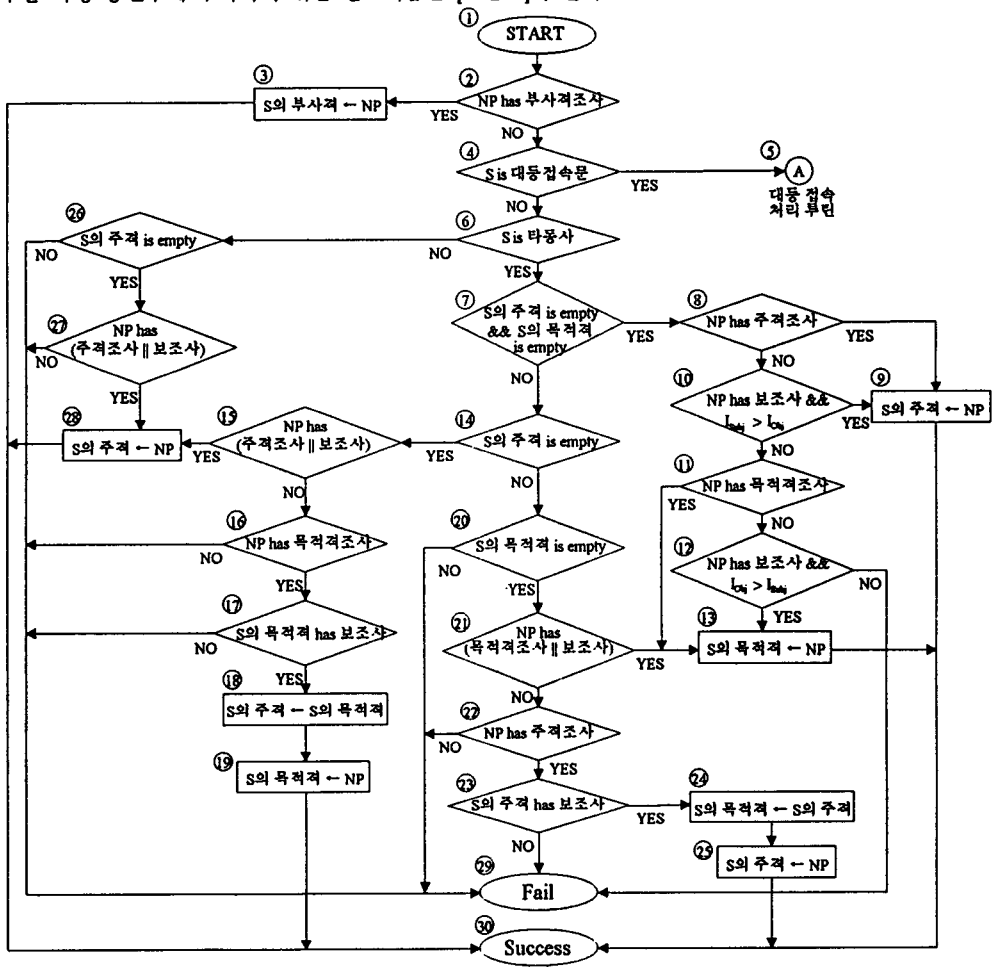


(2)-④

3. 중의성 해소 방법

3.1 명사구의 격결정 알고리즘

NP를 S에 불일 때 (격을 채울 때) 첫째, 불일 수 있는지, 둘째, 불일 수 있으면 어떤 격으로 불는지를 처리 해야 한다. 이것은 규칙 NP-S의 실행에서 수반되는 작업이다(특정 명사구를 특정 용언구에 부착하는 문제). 특정 명사구를 특정 용언구에 부착하기 위한 알고리즘은 [그림 1]과 같다.



[그림 1] 중의성 해소 알고리즘 1

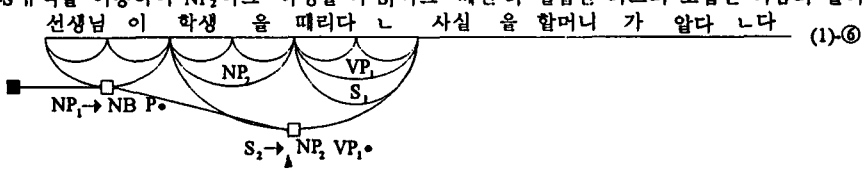
이 알고리즘에서는 상호 정보가 이용된다. 상호 정보 $I_{rel}(x, y)$는 다음과 같이 정의된다[2].

$$I_{rel}(x, y) = \log_2 \frac{P(x, y)}{P(x)P(y)} \approx \log_2 \frac{N\, f_{rel}(x, y)}{f(x)\, f(y)}$$

$$where \quad rel \in \{subj, obj, mod, \cdots\}$$

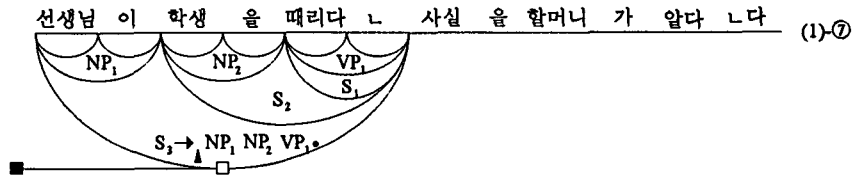
상호 정보(mutual information)는 직관적으로 말하여 x와 y가 실제로 공기한(co-occure) 확률 $P(x, y)$를 x와 y가 우연히 함께 나타날 확률 $P(x)P(y)$로 나눈 것을 의미한다. $f(x)$, $f(y)$는 각각 x와 y가 말뭉치 내에서 나타난 빈도이며 $f_{rel}(x, y)$는 x와 y가 rel이라는 통사적 관계로 함께 나타난 빈도이다. rel의 값으로는 현재 subj, obj, mod가 있다. 이를 조사 단위로 세분할 수도 있다. 알고리즘에서 I_{subj}와 I_{obj}는 $I_{subj}(x, y)$, $I_{obj}(x, y)$를 줄여서 나타낸 것이다. 이 알고리즘에서 x는 NP 내의 N(명사)이고 y는 S 내의 V(용언)이다.

(1)에서 NP-S 규칙을 이용하여 NP_2아크 '학생을'이 S_1아크 '때린'과 결합한 파스의 모습은 다음과 같다.



(1)-⑥

(1)-⑥은 알고리즘에서 1 → 2 → 4 → 6 → 7 → 8 → 10 → 11 → 13 → 30 의 경로를 따른다. NP_2 '학생을'이 목격적 조사를 가지며, S_1의 '때린'이 타동사이며 주격과 목적격이 하나도 채워져 있지 않으므로 '학생을'은 '때린'의 목적격을 채우게 된다. (1)-⑥의 NP_1 '선생님이'와 S_2 '학생을 때린'에 NP-S 규칙을 한번 더 실행하면 다음과 같다.

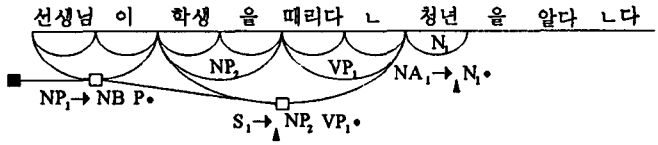


(1)-⑦

(1)-⑦은 알고리즘에서 1 → 2 → 4 → 6 → 7 → 14 → 15 → 28 → 30 의 경로를 따라 분석된다. S_2의 '때린'에 주격이 비어 있고 NP_1 '선생님이'가 주격 조사를 가지므로 '선생님이'는 '때린'의 주격을 채우게 된다. 우리는 이 알고리즘을 이용하여 NP가 S에 연결이 가능한지의 여부를 알 수 있고, NP가 S와 연결 가능할 때 NP의 격을 결정할 수 있다. 알고리즘에서 Fail로 끝나게 되면 NP가 S에 부착되지 못한 상태로 스택에 남게 된다.

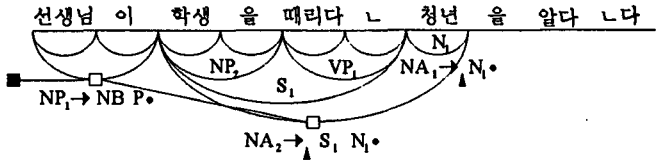
3.2. 피수식 명사의 격결정 알고리즘

관형용언구(S)가 명사(NA)를 수식할 때에는 피수식 명사구(NA)가 S의 어떤 격을 채우는 지 결정해야 한다(특정 관형용언구의 특정 명사구에의 부착 문제). 특정 관형용언구(S)가 특정 명사(NA)를 수식할 때 피수식 명사의 격결정은 [그림 2]의 알고리즘을 따른다. (모습종류: '사실', '것', '모습' 등 용언구의 격을 채우지 않으며 수식을 받는 명사 부류) (2)가 '청년'까지 다음과 같이 분석되었다고 하자.



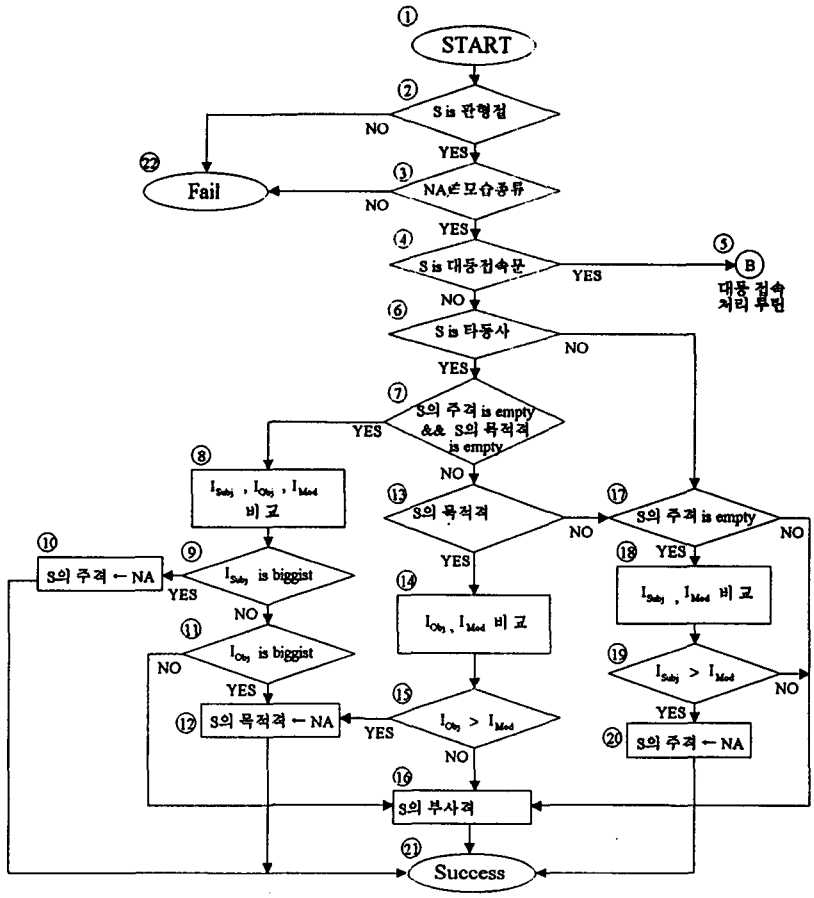
(2)-⑤

다음 S_1아크가 NA_1아크와 결합하여 생긴 새 아크 NA_2는 다음과 같이 된다.



(2)-⑥

(2)-⑥은 위의 알고리즘에서 경로 1 → 2 → 3 → 4 → 6 → 7 → 13 → 17 → 18 → 19 → 20 → 21 을 따른다. S_1의 '때리다'의 목적격을 NP_2가 이미 채우고 있으므로 '청년'은 주격이나 부사격을 채우게 된다. '청년'은 모습종류가 아니므로 용언에 격을 채우게 된다. 알고리즘 상에 18에서 N_1 '청년'이 주격을 채우는 것이 좋은지 부사격을 채우는 것이 좋은지 비교한다. (2)-⑥에서 N_1 '청년'이 '때리다'의 주격과 부사격 중 어떤 격을 채우는 것이 좋은지 알아내기 위해서 상호 정보 I_{subj}(청년, 때리다)와 I_{mod}(청년, 때리다)를 비교한다. I_{subj}(청년, 때리다) > I_{mod}(청년, 때리다)이므로



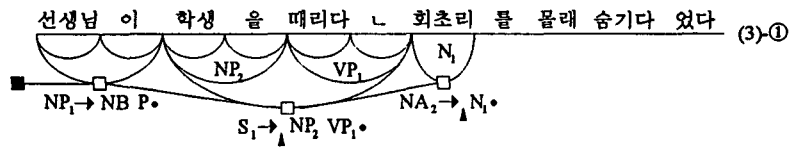
[그림 2] 중의성 해소 알고리즘 2

피수식명사 '청년'은 '때리다'의 주격을 채운다.

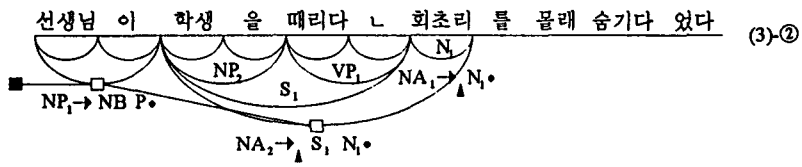
관형용언구가 명사를 수식하는 다른 예를 보자.

(3) 선생님이 학생을 때린 회초리를 몰래 숨겼다

'회초리'가 나온 순간에 파스의 모습은 다음과 같다.



S_1의 용언 '때린'은 타동사이며 NP_2 '학생을'이 목적격을 채우고 있다. 이 때, S_1 아크가 NA_2 아크에 연결되면 챠트는 다음과 같이 된다.



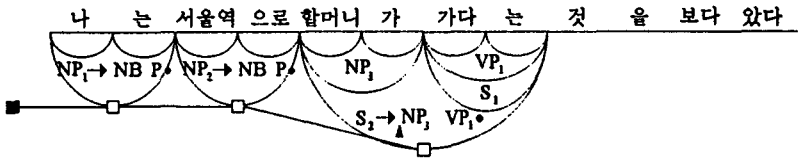
이 때, 피수식 명사의 격결정 알고리즘에서 1 → 2 → 3 → 4 → 6 → 7 → 13 → 17 → 18 → 19 → 16 → 21 의 경로를 따른다. '때리다'의 목적격이 이미 채워져 있으므로 '회초리'는 목적격이 될 수 없다. 알고리즘 상의 18에 '회초리'가 '때리다'의 주격을 채우는 것이 좋은지 부사격을 채우는 것이 좋은지를 결정하기 위한 비교가 일어난다. I_{mod}(회초리, 때리다) > I_{subj}(회초리, 때리다)이므로 '회초리'는 '때리다'의 부사격을 채우게 된다.

3.3. 명사구가 부착될 용언구의 결정

하나의 명사구 뒤에 여러 개의 용언구가 나올때 어느 것과 연결될지 알 수 없다(각 경우가 모두 가능하다). 이를 나타내기 위해서 각 경우마다 파스를 생성한다. 결국 여러 개의 파스가 생기게 된다. 명사구 '서울역으로' 뒤에 두 개의 용언구가 나오는 다음 예를 보자.

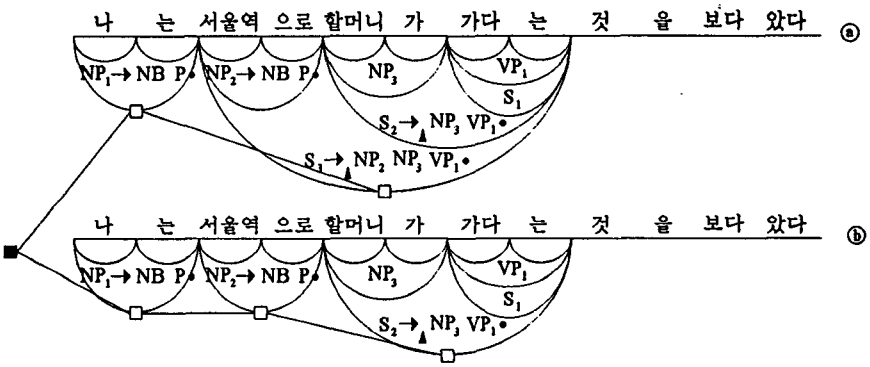
(4) 나는 서울역으로 할머니가 가는 것을 보았다.

(4)의 파싱 과정에서 다음 상태에 다다랐다고 하자.

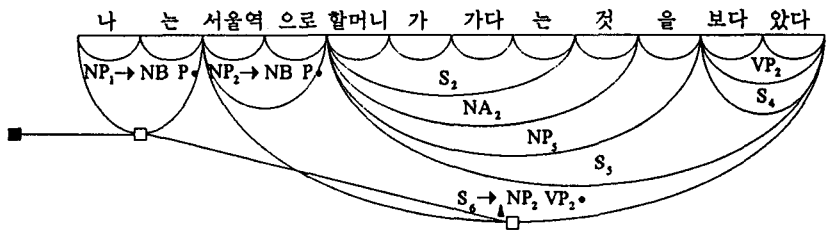


(4)-①

NP_2 아크와 S_2 아크가 인접해 있으므로 다시 NP-S 규칙을 적용할 수 있다. 그러나 NP_2 '서울역으로'는 더 뒤에 나오는 용언구와 결합할 수도 있으므로 아래와 같이 두개의 파스가 생성되어야 한다. ⓐ는 '서울역으로'가 '가는'에 부착된 경우이고 ⓑ는 '서울역으로'가 더 뒤에 나오는 용언구에 부착되기 위해서 남아있는 경우이다.



(4)-②



ⓐ가 맞는지 ⓑ가 맞는지는 현재 시점에서 결정하기 어려우며 따라서 두 가능성을 모두 생성해야 한다(뒤에 나오는 용언의 수만큼 '서울역으로'가 붙어야 하는 중의성이 생긴다). 결국 다중 파스 간의 비교를 통하여 중의성을 해소하여야 한다.

ⓑ가 진행되면 ⓑ'에서 보듯이 NP_2는 결국 '보았다'에 연결된다. '서울역으로'가 '가는'에 연결될지 '보았다'를 연결될지에 대한 중의성을 해소하기 위해 통계적인 방법을 이용한다. NP_2 '서울역으로'는 부사격 조사를 갖는다. 부사격 관계는 시각, 방향, 장소에 따라 의미가 달라지므로 각각을 구분하여 상호 정보를 얻어야 한다. 시각, 방향, 장소에 대한 부사격 관계의 상호 정보를 각각 $I_{mod1}(A, B)$, $I_{mod2}(A, B)$, $I_{mod3}(A, B)$로 나타낸다. '서울역으로'는 방향을 나타내므로 I_{mod2}(서울역, 가다)와 I_{mod2}(서울역, 보다)의 값을 비교하여 결정한다. I_{mod2}(서울역, 가다) > I_{mod2}(서울역, 보다)이므로 ⓐ가 ⓑ를 이기게 되어 ⓑ가 제거된다.

(5) 나는 전망대에서 청년이 사는 집을 보았다.

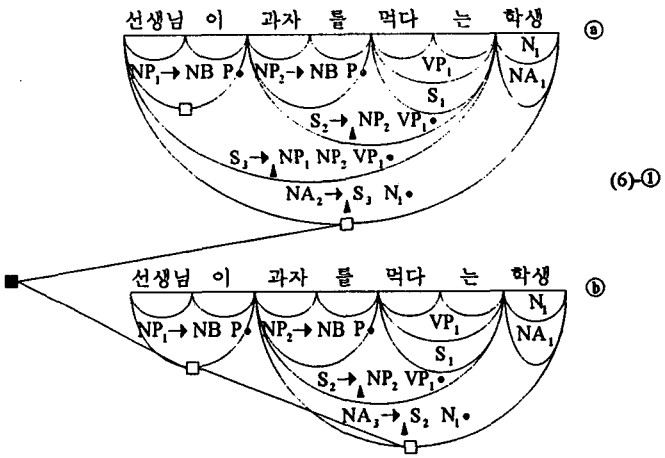
(5)에서도 '전망대에서'가 각각 '사는'과 '보았다'에 연결될 수 있는 중의성이 발생한다. '전망대에서'는 장소를 나타내는 부사격 조사 '에서'를 갖는 부사격 명사구 이므로 I_{mod3}(전망대, 살다), I_{mod3}(전망대, 보다)를 비교한다. I_{mod3}(전망대, 보다) > I_{mod3}(전망대, 살다)이므로 '전망대에서가 '보았다'를 수식하는 파스가 이긴다.

3.4. 관형용언구가 명사를 수식할 때 중의성 해소

관형용언구가 명사를 수식할 때, 여러 개의 관형용언구가 하나의 명사를 수식하거나, 하나의 관형용언구가 여러 다른 명사를 수식하는 중의성이 일어날 수 있다. 이 때 스택상에는 여러 개의 파스가 생성된다. 다음 예에서 피수식 명사의 격결정과 다중 스택 상에서 중의성이 나타난 파스의 비교가 어떻게 일어나는 지를 알아보자.

(6) 선생님이 과자를 먹는 학생 ...

(6)을 분석하면 다음과 같이 두개의 파스가 생기게 된다.



(6)-①에서 피수식 명사 '학생'이 다른 두개의 관형용언구의 수식을 받는 중의성이 생기게 된다. 3.2 절에서 설명한 피수식 명사의 격결정 알고리즘에 따라 (6)-①-ⓐ에서 '학생'은 '먹는'의 부사격을 채우며, (6)-①-ⓑ에서 '학생'는 주격을 채운다. 피수식 명사 '학생'의 격이 각각 다르게 결정되었으므로 상호 정보를 이용하여 피수식 명사 '학생'이 어떤 격으로 쓰이는 것이 적합한지 비교한다. I_{subj}(학생, 먹다) > I_{nmod}(학생, 먹다)이므로 (6)-①-ⓑ가 이긴다.

4. 실 험

우리는 30 만 어절의 코퍼스를 이용하여 100 문장을 실험하였다. 상호 정보의 비교를 하여야 하는 경우에 상호 정보값이 둘다 $-\infty$가 나오는 경우는 알고리즘을 적용하지 못하게 되며 실험 문장내에서 이 경우가 발생하는 비율은 다음과 같다.

	I_1	I_2	비율
①	$-\infty$	$-\infty$	67%
②	other wise		33%

①의 경우는 상호 정보 I_1과 I_2를 구하는 데, 주어진 관계를 갖는 트리플 쌍이 코퍼스내에 나타나지 않는 것이다. 이 경우는 I_1과 I_2가 모두 $-\infty$값을 갖게 되어 상호 정보를 이용한 비교를 할 수 없게 된다. 위의 표를 보면 실험할 수 없는 경우의 비중이 67%를 차지한다. 이는 우리가 이용한 말뭉치의 데이타 양이 적어서 생기는 문제이다. 앞으로 데이타의 양이 많은 말뭉치를 이용하게 되면 ①의 비중은 줄어들 것이다.

I_1과 I_2 둘 중에 하나라도 상호 정보를 구할 수 있는 ②의 경우는 상호 정보를 비교하여 중의성을 해소를 하는 데 이용할 수 있다. 실험을 통해 나온, 상호 정보의 차이에 따른 분석 정확도와 전체에서의 비율을 보면 다음과 같다.

	$\lvert I_1 - I_2 \rvert$	분석 정확도 : X	전체에서의 비율
①	0 ~ 2	64%	12%
②	2 ~ 4	89%	26%
③	4 이상	97%	62%

상호 정보의 차가 클수록 분석 정확도가 높다. 상호 정보의 비교가 가능한 경우에는 상호 정보의 차가 큰 것들이 더 많은 부분을 차지한다. 결국 실험 가능한 경우에 대해서는 문장 분석의 정확도는 약 91%가 된다는 것을 알 수 있다.

5. 결 론

본 논문에서는 한국어의 구조적 중의성 중에서 명사구의 용언구에 대한 부착에서 발생하는 중의성, 관형용언구의 명사구에 대한 수식에서 발생하는 중의성을 해소하는 기법을 다루었다. 이를 위해 ECFG 로 한국어를 표현하였으며 이에 대한 파싱을 위해 기존의 챠트 파서를 확장하였다. 중의성으로 인하여 다중 파스가 발생하게 되면 이를 스택으로 관리하며 적정 시점에서 비교 가능한 두 파스를 통계적인 정보에 의하여 비교하여 하나를 제거하도록 하였다. 명사구와 이것이 부착될 용언구가 주어진 상황에서 그리고 관형용언구와 이의 피수식 명사구가 주어진 상황에서 이에 대한 격 결정을 위한 알고리즘을 제시하였다. 실험 결과 우리의 중의성 해소 기법이 매우 효과적임을 알 수 있었다.

참고 문헌

[1] 김재훈, 서정연, 김길창, 구문 그래프를 이용한 구문적 애매성 분석, 1992 년도 제 4 회 한글 및 한국어 정보처리 학술발표 논문집, pp. 159-167, 1992.

[2] 심광섭, 김영택, "통계 정보를 이용한 구조적 모호성 해소," 정보과학회 논문지, 21 권 2 호, pp.341-349, 1994.2.

[3] 양재형, 김영택, 통계 정보를 활용한 한국어 미지격 명사구의 문법기능 결정, 정보과학회 논문지, 21 권 5 호, pp.808-815, 1994.

[4] 양재형, 김영택, 다중 지식원을 이용한 한국어의 분석, 정보과학회 논문지, 21 권 7 호, pp1324-1332, 1994.

[5] 윤덕호, 김영택, 미지문법관계 속성을 이용한 LFG 에서의 한국어 문장분석 연구, 정보과학회 논문지, 16 권 5 호, pp. 434-444, 1989.

[6] 양성일, 나동렬, "트리합성 문법을 이용한 한국어 파싱," 제 6 회 한글 및 한국어정보처리 학술대회 논문집, pp.426-433, 1994.11.

[7] 서정수, 국어문법, 뿌리깊은나무, 1994.